Modeling and Optimization in Science and Technologies

Xin-She Yang
Gebrail Bekdaş
Sinan Melih Nigdeli *Editors*

Metaheuristics and Optimization in Civil Engineering

Springer

Modeling and Optimization in Science and Technologies

Volume 7

About this Series

The book series *Modeling and Optimization in Science and Technologies (MOST)* publishes basic principles as well as novel theories and methods in the fast-evolving field of modeling and optimization. Topics of interest include, but are not limited to: methods for analysis, design and control of complex systems, networks and machines; methods for analysis, visualization and management of large data sets; use of supercomputers for modeling complex systems; digital signal processing; molecular modeling; and tools and software solutions for different scientific and technological purposes. Special emphasis is given to publications discussing novel theories and practical solutions that, by overcoming the limitations of traditional methods, may successfully address modern scientific challenges, thus promoting scientific and technological progress. The series publishes monographs, contributed volumes and conference proceedings, as well as advanced textbooks. The main targets of the series are graduate students, researchers and professionals working at the forefront of their fields.

More information about this series at http://www.springer.com/series/10577

Xin-She Yang · Gebrail Bekdaş
Sinan Melih Nigdeli
Editors

Metaheuristics and Optimization in Civil Engineering

Springer

Editors
Xin-She Yang
School of Science and Technology
Middlesex University
London
UK

Sinan Melih Nigdeli
Faculty of Engineering
Istanbul University
Istanbul
Turkey

Gebrail Bekdaş
Faculty of Engineering
Istanbul University
Istanbul
Turkey

ISSN 2196-7326 ISSN 2196-7334 (electronic)
Modeling and Optimization in Science and Technologies
ISBN 978-3-319-26243-7 ISBN 978-3-319-26245-1 (eBook)
DOI 10.1007/978-3-319-26245-1

Library of Congress Control Number: 2015954625

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

Almost all design problems in engineering can be considered as optimization problems and thus require optimization techniques to solve. However, as most real-world problems are highly nonlinear, traditional optimization methods usually do not work well. The current trend is to use evolutionary algorithms and metaheuristic optimization methods to tackle such nonlinear optimization problems. Metaheuristic algorithms have gained huge popularity in recent years. These metaheuristic algorithms include genetic algorithms, particle swarm optimization, bat algorithm, cuckoo search, differential evolution, firefly algorithm, harmony search, flower pollination algorithm, ant colony optimization, bee algorithms, and many others. The popularity of nature-inspired metaheuristic algorithms can be attributed to their good characteristics because these algorithms are simple, flexible, efficient, and adaptable, and yet easy to implement. Such advantages make them versatile to deal with a wide range of optimization problems without much a priori knowledge about the problem to be solved.

Metaheuristic algorithms play an important role in the optimum design of complex engineering problems when analytical approaches and traditional methods are not effective for solving nonlinear design problems in civil engineering. Generally speaking, these design problems are highly nonlinear with complex constraints, and thus are also highly multimodal. These design constraints often come from design requirements and security measures such as the stresses on the members due to external loading, environmental factors, and usability under service loads. A mathematical solution may be the best approach in an ideal world, but in engineering designs, the values of a design variable such as mass or length must be realistic; for example, quantities must be nonnegative. In addition, such design values must correspond to something that can be manufacturable in practice.

For all engineering disciplines, optimization is crucially important in the design process so as to find a good balance between economy and security that are the primary goals of designs. Aesthetics and practicability are also important in real-world applications. Civil engineering is probably the oldest engineering discipline and it has always been linked to the construction and realization of

civilization. In fact, optimization may be more relevant in civil engineering than in other engineering disciplines. For example, in designing a non-critical machine part in mechanical engineering, the stresses on the part must not exceed certain limits. If a stronger part is used, it may become too expensive. On the other hand, a weaker part may still be able to make the machine work properly, but in time such weak parts can be worn off or damaged. However, such parts may be easy to be replaced at low costs. If this is the case, machine serviceability can be maintained in practice. But in civil engineering, structural integrity and safety may impose stringent restrictions on the structural members that may not be easily replaced. In such cases, all design constraints and the best possible balance between security and economy must be found without risking lives. In addition, sometimes, the minor improvement may not be as important as robustness in applications. A robust design should be able to handle uncertainties in terms of material properties, manufacturing tolerance, and load irregularity in service. Due to complexity and a large number of design constraints in civil engineering, traditional methods often struggle to cope with such high nonlinearity and multimodality. Thus, metaheuristic optimization methods have become important tools in the optimum design in civil engineering.

This edited book strives to summarize the latest developments in optimization and metaheuristic algorithms with emphasis on applications in civil engineering. Topics include the overview of meteaheuristic algorithms and optimization, structural optimization by flower pollination algorithm, steel design by swarm intelligence, optimum seismic design of steel frames by bat algorithm, 3D truss optimization by genetic algorithms, reactive power optimization by cuckoo search, structural design by harmony search, asphalt pavement management, reinforced concrete beam design, transport infrastructure planning, water distribution networks, capacitated vehicle routing, slope stability problems, and others. Therefore, this timely book can serve as an ideal reference for graduates, lecturers, engineers, and researchers in civil engineering, mechanical engineering, transport and geotechnical engineering. It can also serve as a timely reference for relevant university courses in all disciplines in civil engineering.

We would like to thank the editors and staff at Springer for their help and professionalism. Last but not least, we thank our families for their help and support.

June 2015

Xin-She Yang
Gebrail Bekdaş
Sinan Melih Nigdeli

Contents

Contributors

Alper Akin Thomas & Betts Corporation, Meyer Steel Structures, Memphis, TN, USA

Ashkan Allahyari Nik Department of Civil Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

Ibrahim Aydogdu Department of Civil Engineering, Akdeniz University, Antalya, Turkey

Gebrail Bekdaş Department of Civil Engineering, Istanbul University, Avcılar, Istanbul, Turkey

Omid Bozorg Haddad University of Tehran, Tehran, Iran

Serdar Carbas Department of Civil Engineering, Karamanoglu Mehmetbey University, Karaman, Turkey

Y.M. Cheng Department of Civil and Environmental Engineering, Hong Kong Polytechnic University, Hung Hom, Hong Kong

Paulo A.L.F. Coelho Department of Civil Engineering, University of Coimbra, Coimbra, Portugal

Ana Laura Costa Department of Civil Engineering, University of Coimbra, Coimbra, Portugal

Maria Conceição Cunha Department of Civil Engineering, University of Coimbra, Coimbra, Portugal

Ayşe Turhan Daloğlu Department of Civil Engineering, Karadeniz Technical University, Trabzon, Turkey

S.O. Degertekin Department of Civil Engineering, Dicle University, Diyarbakir, Turkey

Herbert H. Einstein Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, USA

Zong Woo Geem Department of Energy IT, Gachon University, Seongnam, South Korea

Navid Ghajarnia University of Tehran, Tehran, Iran

Saeed Gholizadeh Department of Civil Engineering, Urmia University, Urmia, Iran

S.K. Joshi Department of Electrical Engineering, The M.S. University of Baroda, Vadodara, India

Hugo A. Loáiciga University of California, Santa Barbara, Santa Barbara, CA, USA

Qifang Luo College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, China

Miguel Mariño University of California, Davis, Davis, CA, USA

H.K. Mewada Department of Electronics and Communications, CSPIT, Charotar University of Science and Technoloogy, Changa, India

Fereidoon Moghadas Nejad Department of Civil and Environment Engineering, Amirkabir University of Technology, Tehran, Iran

Sinan Melih Nigdeli Department of Civil Engineering, Istanbul University, Avcılar, Istanbul, Turkey

J.K. Pandya Department of Civil Engineering, Dharmsinh Desai University, Nadiad, India

K.S. Pandya Department of Electrical Engineering, CSPIT, Charotar University of Science and Technology, Changa, India

Hamed Poorhoseini Department of Civil Engineering, Urmia University, Urmia, Iran

Mehmet Polat Saka Department of Civil Engineering, University of Bahrain, Isa Town, Bahrain

Mohammad Solgi University of Tehran, Tehran, Iran

Vedat Toğan Department of Civil Engineering, Karadeniz Technical University, Trabzon, Turkey

Jian Xie College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, China

Xin-She Yang Design Engineering and Mathematics, School of Science and Technology, Middlesex University, The Burroughs, London, UK

H. Zakeri Amirkabir Artificial Intelligence and Image Processing Lab (Attain), Department of Civil and Environment Engineering, Amirkabir University of Technology, Tehran, Iran

Hongqing Zheng College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, China

Yongquan Zhou College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, China; Guangxi High School Key Laboratory of Complex System and Computational Intelligence, Nanning, China

Review and Applications of Metaheuristic Algorithms in Civil Engineering

Xin-She Yang, Gebrail Bekdaş and Sinan Melih Nigdeli

Abstract Many design optimization problems in civil engineering are highly nonlinear and can be challenging to solve using traditional methods. In many cases, metaheurisitc algorithms can be an effective alternative and thus suitable in civil engineering applications. In this chapter, metaheuristic algorithms in civil engineering problems are briefly presented and recent applications are discussed. Two case studies such as the optimization of tuned mass dampers and cost optimization of reinforced concrete beams are analyzed.

Keywords Metaheuristic algorithms · Civil engineering · Optimization

1 Introduction

Metaheuristic algorithms play a great role in the optimum design of complex engineering problems when analytical approaches and traditional methods are not effective for solving nonlinear design problems in civil engineering. Generally speaking, these design problems are highly nonlinear with complex constraints, and thus are also highly multimodal. These design constraints often come from design

X.-S. Yang (✉)
Design Engineering and Mathematics, Middlesex University, The Burroughs, London NW4 4BT, UK
e-mail: x.yang@mdx.ac.uk; xy227@cam.ac.uk

X.-S. Yang
School of Science and Technology, Middlesex University, The Burroughs, London NW4 4BT, UK

G. Bekdaş · S.M. Nigdeli
Department of Civil Engineering, Istanbul University, 34320 Avcılar, Istanbul, Turkey
e-mail: bekdas@istanbul.edu.tr

S.M. Nigdeli
e-mail: melihnig@istanbul.edu.tr

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*, Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_1

requirements and security measures such as stresses on the members due to external loading, environmental factors, and usability under service loads. In addition to the design constraints, solution ranges for the design variables must be defined. A mathematical solution may be the best approach in an ideal world, but in engineering design, if a variable is a mass or a length, the value of the design variable must be realistic, for example, it cannot be negative. In addition, the design values must correspond to something that can be manufacturable in practice.

For all engineering disciplines, optimization is needed in the design in order to find a good balance between economy and security which are the primary goals of designs and engineers should not ignore any of these two goals. Esthetics and practicability are the other important goals which are also important in many specific design problems. Thus, these goals should also be considered in addition to the primary goals, if needed.

Civil engineering is the oldest engineering discipline and it has been linked to one of humanity's most important needs—the construction and realization of civilization. In fact, optimization may be more relevant in civil engineering than in other engineering disciplines. For example, in designing a machine part in mechanical engineering, the stresses on the part must not exceed the security measures. If we produce a stronger part, it will be too expensive. If the part is not strong enough, it can serve, but in time, the part can be damaged. In that case, this part may be replaced with a new one and serviceability of the machine can still be sustained. This process is normal for mechanical engineering designs, but we cannot replace a structural member easily in civil engineering. Also, in civil engineering, big and complex systems are investigated. In that case, we need to consider all design constraints and the best balance between security and economy must be found without risking lives. Due to various design constraints, mathematical optimization may not be effective in civil engineering. Thus, metaheuristic methods are important in the optimum design of civil engineering.

In this chapter, metaheuristic algorithms used in civil engineering will be presented with some literature reviews. In addition, two optimization case studies in applications will be presented in detail. These examples include the optimization of reinforced concrete beams and tuned dampers for the reduction of vibrations.

2 Metaheuristic Algorithms

In engineering, an optimum design problem can be written in mathematical form as

$$\text{Minimize } f_i(x), x \in \mathcal{R}^n, (i = 1, 2, \ldots N) \tag{1}$$

subject to

$$h_j(x), (j = 1, 2, \ldots, J), \tag{2}$$

$$g_k(x) \leq 0, (k = 1, 2, \ldots, K) \tag{3}$$

where

$$\mathrm{x} = (x_1, x_2, \ldots, x_n)^T, (i = 1, 2, \ldots, n) \tag{4}$$

is the design vector containing design variables. The objective functions (f_i (x)), design constraints about equalities (h_j (x)), and inequalities (g_i (x)) are the function of the design vector (x). In an optimization process using metaheuristic algorithms, design variables are randomly assigned and then, the objective functions and design constraints are calculated. Design constraints are generally considered by using a penalized objective function. If a particular set of values of design variables is not suitable for a design constraint, the objective function, which needs to be minimized, should be increased with some penalty.

The set of design variables, or the design vector, is generated several times (or the number of population) and stored as a matrix containing possible solutions. This is the initial part of the algorithm and the process is similar for all metaheuristic algorithms. After this initial process, the aim is to try to improve the results based on the special principles of the algorithm of interest. These principles are different for each metaheuristic algorithm and are often inspired by or related to a biological or natural process. Metaheuristic algorithms inspired from observations of a process usually provide a set of updating equations that can be used to update the existing design variables during iteration.

In this section, several metaheuristic algorithms are summarized and the relevant literature studies concerning these algorithms are also discussed.

2.1 Genetic Algorithm

Genetic algorithm (GA) is one of the oldest metaheuristic algorithms. It is based on Charles Darwin's theory of natural selection. The properties include the crossover and recombination, mutation and selection by Holland [1]. The procedure of GA can be summarized in the following seven steps:

Step 1 The optimization objective is encoded.
Step 2 A fitness function or criterion for selection of an individual is defined.
Step 3 A population of individuals is initialized.
Step 4 The fitness function is evaluated for all individuals.
Step 5 A new population is generated using the rules of natural selection. These rules are crossover, mutation, and proportionate reproduction.

Step 6 The population is evolved until a defined stopping criterion is met.
Step 7 The results are decoded so as to obtain the solutions to the design problem.

The application of GA in civil and structural engineering dates back to 1986, when Goldberg and Samtoni used GA for the optimum design of a 10-bar truss system [2]. Until now, GA and its variants have been successfully employed in optimization of structural engineering problems [3]. Recent applications are structural system identification [4], design of long-span bridges [5], topology optimization of steel space-frame roof structures [6], truss topology optimization [7], and many others. Transportation engineering is also a major application area of GA. Recently, several approaches to urban traffic flow [8], traffic signal coordination problem [9], emergency logistic scheduling [10], and calibration of rail transit assignment models are proposed [11].

2.2 Simulated Annealing

Annealing is a process in materials science. In the annealing process, a metal is heated so that its structure can rearrange during slow cooling so as to increase the ductility and strength of the metal. During such controlled cooling, atoms arrange into a low energy state (crystallized state). If the cooling process is quick, a polycrystalline state occurs which is corresponding to a local minimum energy. In the simulated annealing (SA) algorithm, Kirkpatrick et al. [12] and Cerny [13] used the annealing process as an inspiration.

SA has been used to solve many optimization problems in civil engineering and the recent applications are as follows. Costa et al. employed SA for planning high-speed rail systems [14]. Tong et al. used an improved SA in optimum placement of sensors [15]. A genetic SA algorithm is employed in the collapse optimization for domes under seismic excitation [16]. Server systems were designed with an SA-based procedure by Karovic and Mays [17]. SA was also used in thermal building optimization by Junghans and Darde [18].

2.3 Ant Colony Optimization

Ants live in a colony and the population of their colony is between 2 and 25 millions. They can lay scent chemicals or pheromone as a means to communicate. Each ant follows pheromone trails, and when exploring the sounding, more pheromone will be laid from/to the food source. Their behavior can form some emerging characteristics and the ant colony optimization algorithm was developed by Marco Dorigo in 1992 [19].

Ant colony optimization (ACO) has also been applied for several structural engineering problems [3]. Researchers continue to study new applications of civil engineering problems by employing ACO. Recently, multi-compartment vehicle routing problem [20], traffic engineering problems [21], determination problem of noncircular critical slip surface in slope stability analysis [22], and multiobjective structural optimization problems [23] were solved by ACO.

2.4 Particle Swarm Optimization

In 1995, Kennedy and Ebarhart [24] developed particle swarm optimization (PSO) which imitates the behavior of social swarms such as ant colonies, bees, and bird flocks. PSO is a population-based metaheuristic algorithm. In a swarm, particles are randomly generated and new solutions are updated in an iterative manner. The solution particles tend to move toward the current best location, while they move to new locations. Since all particles tend to be the current best solution, the effectiveness of population-based algorithm can be easily recognized if the best solution is an approach to the true global optimality. Compared to GA, PSO uses real-number strings and encoding or decoding of the parameters into binary strings is not needed.

Swarm intelligence applications of structural design [25] and several civil engineering applications [26] were recently presented. Most recent applications are the design of tall buildings [27], size optimization of trusses [28], slope stability analyzing [29], and water distribution systems [30].

2.5 Harmony Search

Harmony search (HS) algorithm is a music-based metaheuristic algorithm. It was developed by Geem et al. [31] after observation of a musician's performance. Musicians search the best harmony by playing harmonic music pieces. Similarly, the objective function of an engineering problem can be considered as a harmony. In a musical performance, the musician plays notes and may modify these notes when needed. The new notes may be similar to a favorite note or a new song. The major application of HS in civil engineering was presented by Yoo et al. [32] and others.

2.6 Firefly Algorithm

The flashing characteristic of fireflies has inspired a new metaheuristic algorithm. Yang [33] developed the firefly algorithm (FA) using the special rules of fireflies. These rules are given as below.

- All fireflies are unisex. Thus, a firefly will be attracted to other fireflies.
- The brightness is related to attractiveness. In that case, the less bright firefly will move toward the brighter one. Attractiveness and brightness will decrease when the distance increases. If there is no brighter one, the firefly will move randomly.
- The landscape of the optimization objective affects and determines the brightness of individuals.

In order to improve the robustness of FA, chaotic maps were included in FA by Gandomi et al. [34]. Since FA is a multimodal algorithm [35], it is suitable for structural optimization [36]. FA has been employed in structural engineering designs such as tower structures [37], continuously cost steel slabs [38], and truss structures [39]. Liu et al. developed a new path planning method using FA in transportation engineering [40].

2.7 *Cuckoo Search*

Yang and Deb [41] developed Cuckoo Search (CS) by idealizing the features of brood parasitism of some cuckoo species as three rules. The first rule is the process in which each cuckoo lays an egg and dumps it in a randomly chosen nest. For the search rule, the best nest with high quality eggs will be carried over the next generations. The number of host nests is fixed. The eggs of a cuckoo may be discovered by the host bird with a probability between 0 and 1 for the last rule.

In structural optimization, CS has been employed in several problems [42]. Also, CS-based design methodologies have been developed for the optimum design of steel frames [43] and truss structures [44]. Ouaarab et al. proposed CS algorithm for the travelling salesman problem [45].

2.8 *Bat Algorithm*

Yang [46] also developed the bat algorithm (BA) by idealizing the echolocation behavior of microbats. Bats fly with varying frequencies, loudness, and pulse emission rates, which can be used to design updating equations of the bat algorithm. This population-based metaheuristic algorithm has been applied to structural optimization problems by Yang and Gandomi [47]. Gandomi et al. investigated constrained problems in structural engineering [48]. Gholizadeh and Shahrezaei investigated the optimum placement of steel plate shear walls for steel structures and employed BA in their optimization method [49]. The BA-based method was developed by Kaveh and Zakian [50] for the optimum design of structures.

Talatahari and Kaveh used an improved BA in optimum design of trusses [51]. Bekdaş et al. optimized reinforced concrete beams by employing BA [52]. Zhou et al. used a hybrid BA with Path Relinking in order to solve capacitated vehicle routing problem [53]. The operations of reservoir systems were optimized by a developed methodology using BA [54].

2.9 Recent Metaheuristic Algorithms

Big bang big crunch (BB-BC) algorithm is a metaheuristic algorithm inspired by the evolution of the universe and developed by Erol and Eksin [55]. In civil engineering, BB-BC algorithm has been employed for truss structures [56–59], steel frame structures [60], parameter estimation of structures [61], and retaining walls [62].

In 2010, Kaveh and Talatahai introduced the Charged System Search (CSS), a metaheuristic algorithm inspired from electrostatic and Newtonian mechanic laws [63]. Recently, CSS has been applied for civil engineering problems such as damage detection in skeletal structures [64], cost optimization of castellated beams [65], optimum design of engineering structures [66, 67], tuned mass dampers [68], and semi-active tuned mass dampers [69].

Krill herd (KH) algorithm is also a metaheuristic algorithm used for structural engineering problems. [70].

Refraction of lights is also used in generation of a metaheuristic algorithm called ray optimization [71]. Kaveh and Khayatazad employed ray optimization for size and shape optimization of truss structures [72]. In transportation engineering Esmaeili et al. used ray optimization in designing granular layers for railway tracks [73].

Also, a newly developed metaheuristic algorithm called flower pollination algorithm is very suitable for engineering problems [74]. This algorithm is presented in a chapter of this book with the topic engineering applications.

3 Optimum Design Examples

In this section, two civil engineering design problems are presented and the results were obtained using metaheuristic methods. In the first example, HS is employed in a multiobjective tuned mass damper methodology. For the second example, BA and HS-based reinforced concrete beam optimum design methodologies are given.

3.1 Example 1: A Multiobjective Optimization of Tuned Mass Dampers for Structures Excited by Earthquakes

Tuned mass damper (TMD), which consists of a mass connected to mechanical system with stiffness and damping elements, has passive vibration absorbers. These devices may be used for damping of vibrations of mechanical systems under random excitations. The performance of the device is dependent on the properties of TMD.

The possible first study on the optimum design of TMDs was proposed by Den Hartog for undamped single degree of freedom (SDOF) main systems [75]. The expressions proposed by Den Hartog for harmonic excitations are still used in practice, including the multiple degrees of freedom structures because Warburton and Ayorinde showed that the structure may be taken as an equivalent SDOF system if the natural frequencies are well separated [76]. For harmonic and random excitations, Warburton derived simple expressions for frequency and damping ratio of TMDs [77].

The simple expressions cannot be derived when damping is included in the main system. Thus, Sadek et al. conducted numerical trials and obtained several expressions using a curve fitting technique. Also, a modification for the expressions was proposed for multiple degrees of freedom (MDOF) structures [78]. Rana and Soong employed numerical optimization for tuned mass dampers in control of a single structural mode and proposed multi-tuned mass dampers for possible control of multiple modes [79]. Chang obtained closed-form expressions for TMDs under wind and earthquake excitations [80]. By investigating the displacement and acceleration response spectra of structures, an extended random decrement method was proposed in the reduction of vibration responses [81]. Alternatively, semi-active magnetorheological (MR) dampers were employed in the design of TMDs by Aldemir [82]. In order to reduce the performance index value, Lee et al. developed a numerical optimization approach for TMDs [83]. Bakre and Jangid proposed mathematical expressions for TMD optimization using numerical searches [84].

Metaheuristic methods have been also used in optimization of TMDs positioned on structures. The metaheuristic methods used in these optimization problems were GA [85–89], PSO [90, 91], bionic optimization [92], HS [93–96], ACO [97], artificial bee optimization [98], shuffled complex evolution [99], and CSS [68].

A shear building with a TMD is physically modeled in Fig. 1. The number of stories of the structure is N. The equations of motion of the structure can be written as

$$\mathbf{M}\ddot{x}(t) + \mathbf{C}\dot{x}(t) + \mathbf{K}x(t) = -\mathbf{M}\{\mathbf{1}\}\ddot{x}_g(t) \tag{5}$$

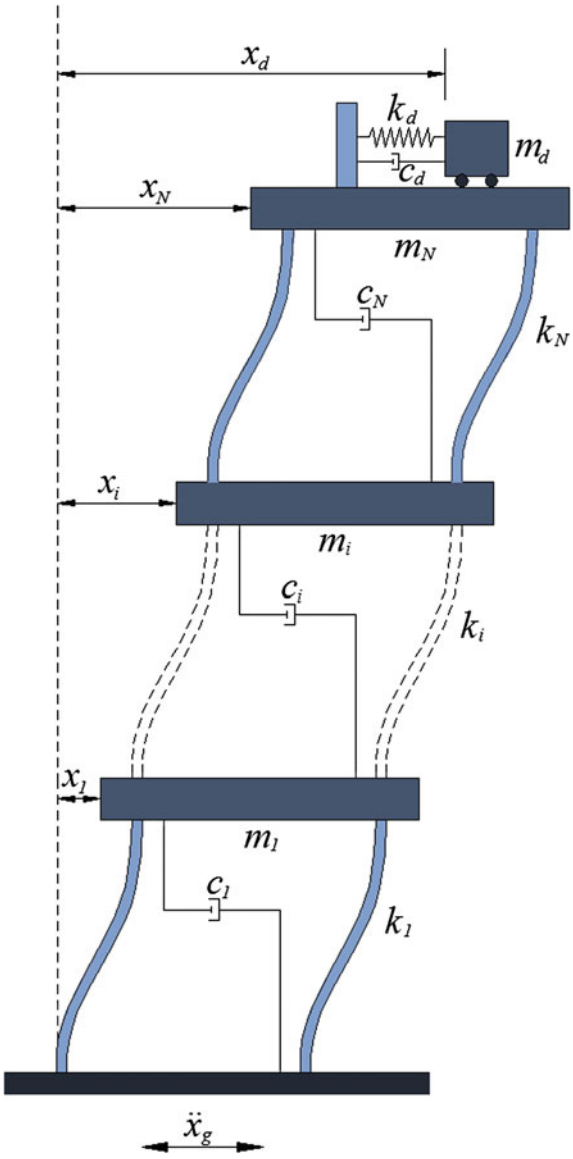

Fig. 1 Physical model of N-story shear building including a TMD

in matrix form for ground acceleration excitations. The M, C, and K matrices are diagonal lumped mass, damping, and stiffness matrices, respectively, and these matrices are given in Eqs. (6)–(8). The $x(t)$, $\ddot{x_g}(t)$ and $\{1\}$ are the vectors containing structural displacements of all stories and TMD (Eq. (9)), ground acceleration in horizontal direction and a vector of ones with a dimension of $(N + 1, 1)$, respectively.

$$M = \text{diag}[m_1 m_2 \ldots m_N m_d] \tag{6}$$

$$C = \begin{bmatrix} (c_1 + c_2) & -c_2 & & & & & \\ -c_2 & (c_2 + c_3) & -c_3 & & & & \\ & \cdot & \cdot & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & & -c_N & (c_N + c_d) & -c_d \\ & & & & & -c_d & c_d \end{bmatrix} \tag{7}$$

$$K = \begin{bmatrix} (k_1 + k_2) & -k_2 & & & & & \\ -k_2 & (k_2 + k_3) & -k_3 & & & & \\ & \cdot & \cdot & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & & -k_N & (k_N + k_d) & -k_d \\ & & & & & -k_d & k_d \end{bmatrix} \tag{8}$$

$$\boldsymbol{x}(t) = [x_1\ x_2 \ldots x_N\ x_d]^{\mathrm{T}} \tag{9}$$

In the equations, m_i, c_i, k_i and x_i are mass, damping coefficient, stiffness coefficient, and displacement of ith story of structure. The parameters of the TMD are shown with mass (m_d), damping coefficient (c_d) and stiffness coefficient (k_d). The displacement of the TMD is x_d. The period (T_d) and damping ratio (ξ_d) of TMD are shown in Eqs. (10) and (11).

$$T_d = 2\pi\sqrt{\frac{m_d}{k_d}} \tag{10}$$

$$\xi_d = 2c_d m_d \sqrt{\frac{k_d}{m_d}} \tag{11}$$

The multiobjective optimization methodology contains two stages: initial calculations and iterative optimization.

At the start of the methodology, optimization constants such as structural properties, external excitations, and ranges of design variables are defined. Then the structure is analyzed by solving the differential equation given as Eq. (5), where we do not know the properties of TMD. These analyses are done for structure without TMD and we need these analyses results in order to use in the objective function. This equation must be solved using numerical iterative analyses because the earthquake excitation has random characteristic and cannot be formulized. A computer code must be developed for the analyses. For the last step of initial calculations, we need to generate possible design variables in order to conduct the iterative optimization. In HS, the initial Harmony Memory (HM) matrix containing

Harmony Vectors (HVs) is generated. The number of HVs is defined with an optimization parameter called Harmony Memory Size (HMS). HVs contain design variables such as mass (m_d), period (T_d) and damping ratio (ξ_d). These design variables are randomly defined. For all set of design variables (each HV), the optimization objectives are calculated. An HM matrix with HVs from 1 to Harmony Memory Size (HMS) and an HV containing the design variables are shown in Eqs. (12) and (13), respectively.

$$\mathrm{HM} = [\,\mathrm{HV}_1 \quad \mathrm{HV}_2 \quad \ldots \quad \mathrm{HV}_{\mathrm{HMS}}\,] \tag{12}$$

$$\mathbf{HV} = \begin{bmatrix} m_{di} \\ T_{di} \\ \xi_{di} \end{bmatrix}. \tag{13}$$

Such multiobjective optimization has two different objectives given in Eqs. (14) and (15). The first objective is the reduction of maximum top story displacement of the structure to a user defined value ($x_{\max}$). If $x_{\max}$ is not a physical value for reduction of displacement for the selected design variable ranges, this value is increased after several iterations. The second objective is used in order to consider the stroke capacity of the TMD, which is now essentially converted into a constraint

$$|\,x_N\,| \leq x_{\max} \tag{14}$$

$$\frac{\max[|x_{N+1} - x_N|]_{\text{with TMD}}}{\max[|x_N|]_{\text{without TMD}}} \leq \text{st\_max} \tag{15}$$

The iterative optimization process starts with generation of a new HV. If the solution of a new vector is better than existing ones in HM, HM is updated by eliminating the worst one. Since the optimization process is multiobjective, the objective given as Eq. (15) is used in elimination. If this objective function is lower than st_max, the other function given in Eq. (14) is considered and the main purpose of the optimization is to minimalize the displacement of the structure without exceeding the stroke capacity. This iterative search is done using the rules of HS and it is finished when the criteria given by two objectives are provided.

According to HS, a new HV is constructed in two ways. With a possibility called Harmony Memory Considering Rate (HMCR), HV is generated by a smaller range and this range is taken around an existing vector in HM. The ratio of the small and whole range is defined with an algorithm parameter called pitch adjusting rate (PAR). If an existing HV is not used as a source, random generation of design variables is done as the generation of initial vectors.

A ten-story structure was optimized as a numerical example [87]. The mass, stiffness coefficient and damping coefficient of a story is 360 t, 6.2 MNs/m, and 650 MN/m, respectively. In optimization, the best value for a structure excited under 44 different earthquake excitations is searched. FEMA P-695 [100] far-fault ground motion set was used and the details of the records of this set are given in

Table 1 FEMA P-695 far-field ground motion records [100]

Earthquake number	Date	Name	Component 1	Component 2
1	1994	Northridge	NORTHR/MUL009	NORTHR/MUL279
2	1994	Northridge	NORTHR/LOS000	NORTHR/LOS270
3	1999	Duzce, Turkey	DUZCE/BOL000	DUZCE/BOL090
4	1999	Hector Mine	HECTOR/HEC000	HECTOR/HEC090
5	1979	Imperial Valley	IMPVALL/H-DLT262	IMPVALL/H-DLT352
6	1979	Imperial Valley	IMPVALL/H-E11140	IMPVALL/H-E11230
7	1995	Kobe, Japan	KOBE/NIS000	KOBE/NIS090
8	1995	Kobe, Japan	KOBE/SHI000	KOBE/SHI090
9	1999	Kocaeli, Turkey	KOCAELI/DZC180	KOCAELI/DZC270
10	1999	Kocaeli, Turkey	KOCAELI/ARC000	KOCAELI/ARC090
11	1992	Landers	LANDERS/YER270	LANDERS/YER360
12	1992	Landers	LANDERS/CLW-LN	LANDERS/CLW-TR
13	1989	Loma Prieta	LOMAP/CAP000	LOMAP/CAP090
14	1989	Loma Prieta	LOMAP/G03000	LOMAP/G03090
15	1990	Manjil, Iran	MANJIL/ABBAR–L	MANJIL/ABBAR–T
16	1987	Superstition Hills	SUPERST/B-ICC000	SUPERST/B-ICC090
17	1987	Superstition Hills	SUPERST/B-POE270	SUPERST/B-POE360
18	1992	Cape Mendocino	CAPEMEND/RIO270	CAPEMEND/RIO360
19	1999	Chi-Chi, Taiwan	CHICHI/CHY101-E	CHICHI/CHY101-N
20	1999	Chi-Chi, Taiwan	CHICHI/TCU045-E	CHICHI/TCU045-N
21	1971	San Fernando	SFERN/PEL090	SFERN/PEL180
22	1976	Friuli, Italy	FRIULI/A-TMZ000	FRIULI/A-TMZ270

Table 2 The ranges of design variables and optimum values

Design variable	Range definition	Optimum values
Mass (t)	Between 1 and 5 % total mass of structure	178.53
Period (s)	Between 0.5 and 1.5 times of the critical period of structure	0.9010
Damping ratio (%)	Between 0.1 and 30 %	29.53

Table 1. The earthquake records were downloaded from Pacific Earthquake Research Centre (PEER) database [101]. In Table 2, the ranges for the design variables and optimum TMD parameters are shown. The st_max was taken as 1 and $x_{\max}$ was taken as zero in order to find a solution minimizing the displacement.

The maximum displacement, total acceleration values, and the scaled maximum TMD displacement (x_d') are given in Table 3 for all excitations. The most critical

Table 3 Maximum responses under FEMA P-695 far-field ground motion records

Earthquake number	Component	Max. (x) (m)		Max. ($\ddot{x} + \ddot{x_g}$) (m/s^2)		x_d'
		Without TMD	With TMD	Without TMD	With TMD	
1	1	0.37	0.30	15.80	11.01	0.94
	2	0.31	0.30	12.99	10.95	1.11
2	1	0.13	0.11	6.33	5.09	0.76
	2	0.22	0.18	9.21	7.22	0.95
3	1	0.26	0.19	12.79	9.81	0.87
	2	0.41	0.32	19.29	14.32	0.97
4	1	0.11	0.14	5.04	5.52	1.34
	2	0.13	0.14	5.46	5.27	1.16
5	1	0.11	0.07	5.33	3.23	0.71
	2	0.19	0.12	7.90	4.99	0.74
6	1	0.08	0.07	4.58	4.42	1.10
	2	0.07	0.09	4.41	5.15	1.14
7	1	0.11	0.10	5.91	4.93	0.93
	2	0.10	0.09	5.12	4.95	0.90
8	1	0.10	0.13	5.00	5.39	1.31
	2	0.08	0.08	3.27	3.09	1.20
9	1	0.15	0.14	8.44	7.25	0.96
	2	0.22	0.20	9.81	9.77	1.06
10	1	0.04	0.04	2.07	2.01	1.05
	2	0.04	0.03	1.99	1.60	0.94
11	1	0.18	0.14	7.42	5.12	0.86
	2	0.11	0.10	5.00	3.91	0.93
12	1	0.08	0.06	6.03	4.29	0.81
	2	0.14	0.12	6.14	5.42	0.95
13	1	0.15	0.16	8.95	6.84	1.11
	2	0.09	0.09	5.01	5.21	1.14
14	1	0.11	0.08	6.68	6.25	0.76
	2	0.12	0.13	6.08	5.90	1.08
15	1	0.12	0.10	6.06	5.21	0.89
	2	0.18	0.16	9.95	7.72	0.88
16	1	0.08	0.13	5.53	5.72	1.49
	2	0.08	0.09	3.35	2.82	1.05
17	1	0.12	0.11	5.11	4.82	1.16
	2	0.14	0.09	6.21	4.55	0.76
18	1	0.18	0.17	8.52	7.95	1.08
	2	0.14	0.13	7.70	6.76	1.05
19	1	0.16	0.13	7.67	5.69	0.99
	2	0.35	0.24	13.83	10.30	0.71

(continued)

Table 3 (continued)

Earthquake number	Component	Max. (x) (m)		Max. ($\ddot{x} + \ddot{x_g}$) (m/s^2)		x_d'
		Without TMD	With TMD	Without TMD	With TMD	
20	1	0.11	0.08	6.65	5.38	0.76
	2	0.15	0.13	7.17	6.75	1.06
21	1	0.09	0.07	4.51	3.86	0.99
	2	0.06	0.04	2.81	1.83	0.68
22	1	0.08	0.07	5.38	4.42	0.99
	2	0.10	0.09	5.27	4.82	1.02

excitation is the second component of the Duzce record. It must be noted that the stroke objective is applied only for the critical excitation. For other excitations, the scaled displacement value may exceed the limitation defined by st_max, but the real displacement of TMD (x_d) is always lower than the result for critical excitation.

The maximum displacement is reduced from 0.41 to 0.32 m for the critical excitation and the top story displacement and acceleration for the critical excitation are plotted in Fig. 2. Also, a steady-state response is observed from the plots. According to the results, an optimum TMD solution is found and the reduction of displacement and accelerations are excellent.

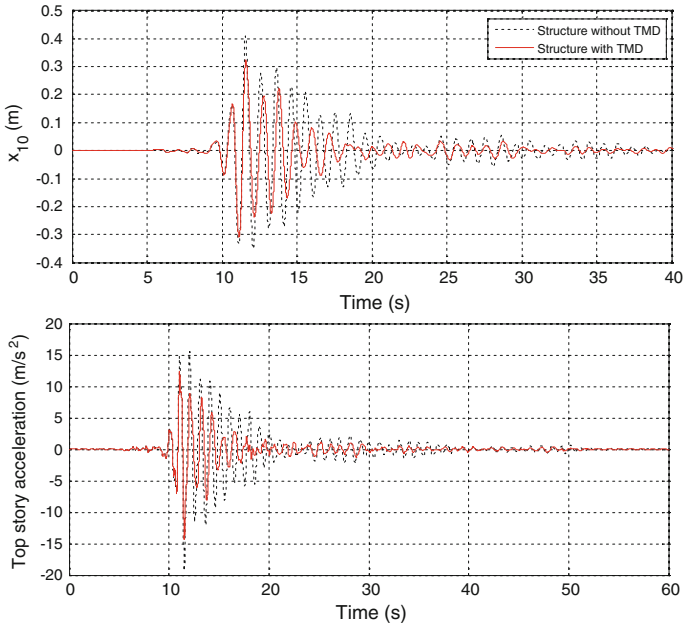

Fig. 2 The time history plots for critical excitation

3.2 Example 2: Optimum Design of Reinforced Concrete Beams

The design of reinforced concrete (RC) members is a challenging task in order to maintain designs with minimum costs. In design of RC members, the experience of the design engineers is important in the best design at minimum cost. The analyses of RC members contain two stages: assume a cross-section and calculate the required reinforcement. We need optimization since an assumption is needed and also, the required reinforcement area cannot be exactly provided by using steel bars in markets with constant/fixed sizes. Concrete and steel have different mechanical behavior. Also, these materials differ in price. Generally, steel is expensive but we need to use it for tensile stresses. By using numerical search, metaheuristic methods such as genetic algorithm [102–106], charged system search [107], harmony search [108, 109], simulated annealing [110], and big bang big crunch [111] have been employed for optimization of RC members.

In this section, two metaheuristic algorithms are separately employed for optimum designs of an RC beam subjected to flexural moment. In the optimum design, the cross-section design and reinforcement design are done by considering the number and size of reinforcements. The optimization objective is to estimate the best design at minimum cost. The design procedures of ACI 318—Building Code Requirements for Structural Concrete [112] have been carried out during the optimization process.

Two algorithms have been used, harmony search (HS) and bat algorithm (BA). First, design constants, the ranges of design variables, and the specific algorithm parameters are defined. The parameters of HS are harmony memory size (HMS), harmony memory considering rate (HMCR), and pitch adjusting rate (PAR). The bat population (n), limits of the pulse frequency ($f_{\min}$ and $f_{\max}$), the pulse rate (r_i), and loudness (A_i) are the parameters of BA.

In the HS approach, a randomization procedure is carried out in order to generate an initial harmony memory matrix containing harmony vectors with possible design variables. Similarly in the BA, displacement vectors are constructed and stored in a matrix. The parameters representing the number of harmony and displacement vectors are defined by HMS and n, respectively. After the generation of design variables, the analyses of RC design are done and the flexural moment capacities are compared with the required flexural moment. Also, steel reinforcements are randomized and rules about the positioning are checked according to the ACI-318 design code. When the initial matrix is generated, the stopping criterion or criteria are checked. If criterion or criteria are not satisfied, the possible solution matrix is modified according to the rules of the algorithms.

As explained before, a new solution vector is modified by generating random numbers from the whole solution range or a range defined around an existing one in HS algorithm and the vector with the worst solution is eliminated from the harmony memory matrix. Differently in the BA, all vectors of the solution matrix are modified by using velocities calculated by random frequencies between $f_{\min}$ and

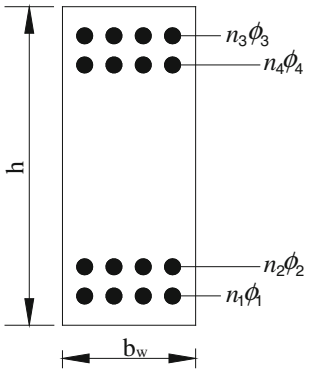

Fig. 3 Design variables of RC beam

$f_{\max}$. The modified displacement vectors are accepted according to the criterion of the pulse rate and loudness. If the pulse rate is smaller than a randomly drawn number, displacement vectors are generated using the initial solution range. Otherwise, the modified solution matrix is accepted. According to BA, the value of the pulse rate is increased while the value of loudness is reduced.

As a numerical example, the optimum design of RC beams was investigated for flexural moments between 50 and 500 kNm. The design variables are shown in Fig. 3, which are breadth (b_w), height (h), number (n_1–n_4), and size (ϕ_1–ϕ_4) of the reinforcements positioned in two lines of compressive and tensile sections. The design constants of the problem are shown in Table 4.

The optimization objective is to minimize the material cost of the beam per unit meter. When the maximum difference in the cost of the best five designs is lower than 2 %, the optimization process is terminated.

The optimum results of the design variables are presented in Tables 5 and 6 for HS and BA-based methods, respectively. BA is more effective than the HS

Table 4 Design constants of RC beam

Definition	Symbol	Unit	Value
Range of the breadth	b_w	mm	250–350
Range of the height	H	mm	350–500
Clear cover	c_c	mm	35
Range of main reinforcement	Φ	mm	10–30
Diameter of stirrup	ϕ_v	mm	10
Max. aggregate diameter	$D_{\max}$	mm	16
Yield strength of steel	f_y	MPa	420
Comp. strength of concrete	f'_c	MPa	20
Elasticity modulus of steel	E_s	MPa	200,000
Specific gravity of steel	γ_s	t/m^3	7.86
Specific gravity of concrete	γ_c	t/m^3	2.5
Cost of the concrete per m^3	C_c	$	40
Cost of the steel per ton	C_s	$	400

Table 5 Optimum results for HS approach

Objective flexural moment (kNm)	50	100	150	200	250	300	350	400	450	500
h (mm)	350	450	500	500	500	500	500	500	500	500
b_w (mm)	250	250	250	250	250	250	300	350	350	350
ϕ_1 (mm)	10	14	18	18	20	28	30	24	30	26
ϕ_3 (mm)	26	26	30	12	12	12	22	10	16	22
n_1	4	4	3	3	4	3	3	5	4	5
n_3	0	0	0	0	0	2	2	3	3	3
ϕ_2 (mm)	12	10	10	22	24	14	12	20	14	16
ϕ_4 (mm)	30	24	16	10	18	16	14	12	10	14
n_2	2	2	4	2	2	4	4	3	4	6
n_4	0	0	0	0	0	0	0	2	4	0
M_u (kNm)	58.61	114.35	172.61	222.87	289.16	334.85	390.41	452.99	502.06	556.40
Cost ($/m)	5.18	6.90	8.34	9.73	11.71	13.35	16.35	18.38	20.54	22.52

Table 6 Optimum results for HS approach

Objective flexural moment (kNm)	50	100	150	200	250	300	350	400	450	500
h (mm)	350	400	500	500	500	500	500	500	500	500
b_w (mm)	250	250	250	250	250	300	300	350	350	350
ϕ_1 (mm)	14	16	16	28	26	22	26	26	28	28
ϕ_3 (mm)	30	26	26	24	14	12	16	12	16	30
n_1	2	3	4	2	3	5	4	5	5	5
n_3	0	0	0	0	2	2	3	6	5	2
ϕ_2 (mm)	12	10	12	10	12	14	12	10	10	16
ϕ_4 (mm)	12	10	20	26	16	26	20	16	16	22
n_2	2	4	2	3	2	2	4	3	2	3
n_4	0	0	0	0	0	0	0	0	0	0
M_u (kNm)	57.31	111.22	167.71	222.42	279.29	333.48	388.94	445.15	500.11	556.17
Cost ($/m)	5.16	6.85	8.20	9.55	11.60	13.56	15.87	18.08	20.16	22.82

approach in minimizing the cost. According to results, doubly reinforcement design is needed for flexural moments higher than 300 kNm.

4 Conclusion

There are many metaheuristic algorithms that can be effective to solve design optimization problems in engineering. This chapter has reviewed some of the most widely used metaheuristic algorithms in the current literature, which include genetic algorithms, bat algorithm, harmony search, ant colony optimization, cuckoo search, firefly algorithm, particle swarm optimization, simulated annealing, and others. Two case studies were also presented with detailed formulations of the problem and some promising results. All these can be thought as a timely snapshot of the vast, expanding literature concerning design optimization in civil engineering. It is hoped that this book may inspire more research in these areas.

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan, Ann Arbor (1975)
2. Goldberg, D.E., Samtani, M.P.: Engineering optimization via genetic algorithm. In: Proceedings of Ninth Conference on Electronic Computation. ASCE, New York, NY, pp. 471–482 (1986)
3. Sahab, M.G., Toropov, V.V., Gandomi, A.H.: A review on traditional and modern structural optimization: problems and techniques. In: Metaheuristic Applications in Structures and Infrastructures, pp. 25–47. Elsevier, Oxford (2013)
4. Marano, G.C., Quaranta, G., Monti, G.: Modified genetic algorithm for the dynamic identification of structural systems using incomplete measurements. Comput. Aided Civil Infrastruct. Eng. **26**(2), 92–110 (2011)
5. Sgambi, L., Gkoumas, K., Bontempi, F.: Genetic algorithms for the dependability assurance in the design of a long-span suspension bridge. Comput. Aided Civil Infrastruct. Eng. **27**(9), 655–675 (2012)
6. Kociecki, M., Adeli, H.: Two-phase genetic algorithm for topology optimization of free-form steel space-frame roof structures with complex curvatures. Eng. Appl. Artif. Intell. **32**, 218–227 (2014)
7. Li, J.P.: Truss topology optimization using an improved species-conserving genetic algorithm. Eng. Optim. **47**(1), 107–128 (2015)
8. Dezani, H., Bassi, R.D., Marranghello, N., Gomes, L., Damiani, F., da Silva, I.N.: Optimizing urban traffic flow using Genetic Algorithm with Petri net analysis as fitness function. Neuro Comput. **124**, 162–167 (2014)
9. Putha, R., Quadrifoglio, L., Zechman, E.: Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions. Comput. Aided Civil Infrastruct. Eng. **27**(1), 14–28 (2012)
10. Chang, F.S., Wu, J.S., Lee, C.N., Shen, H.C.: Greedy-search-based multi-objective genetic algorithm for emergency logistics scheduling. Expert Syst. Appl. **41**(6), 2947–2956 (2014)

11. Zhu, W., Hu, H., Huang, Z.: Calibrating rail transit assignment models with Genetic Algorithm and automated fare collection data. Comput. Aided Civil Infrastruct. Eng. **29**(7), 518–530 (2014)
12. Kirkpatrick, S., GelattJr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
13. Černý, V.: Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. J. Optim. Theory Appl. **45**(1), 41–51 (1985)
14. Costa, A.L., Cunha, M.D.C., Coelho, P.A., Einstein, H.H.: Solving high-speed rail planning with the simulated annealing algorithm. J. Transp. Eng. **139**(6), 635–642 (2013)
15. Tong, K.H., Bakhary, N., Kueh, A.B.H., Yassin, A.Y.: Optimal sensor placement for mode shapes using improved simulated annealing. Smart Struct. Syst. **13**(3), 389–406 (2014)
16. Liu, W., Ye, J.: Collapse optimization for domes under earthquake using a genetic simulated annealing algorithm. J. Constr. Steel Res. **97**, 59–68 (2014)
17. Karovic, O., Mays, L.W.: Sewer system design using simulated annealing in excel. Water Resour. Manage. **28**(13), 4551–4565 (2014)
18. Junghans, L., Darde, N.: Hybrid single objective genetic algorithm coupled with the simulated annealing optimization method for building optimization. Energy Build. **86**, 651–662 (2015)
19. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B **26**, 29–41 (1996)
20. Reed, M., Yiannakou, A., Evering, R.: An ant colony algorithm for the multi-compartment vehicle routing problem. Appl. Soft Comput. **15**, 169–176 (2014)
21. Dias, J.C., Machado, P., Silva, D.C., Abreu, P.H.: An inverted ant colony optimization approach to traffic. Eng. Appl. Artif. Intell. **36**, 122–133 (2014)
22. Gao, W.: Determination of the noncircular critical slip surface in slope stability analysis by meeting ant colony optimization. J. Comput. Civil Eng. (2015)
23. Angelo, J.S., Bernardino, H.S., Barbosa, H.J.: Ant colony approaches for multiobjective structural optimization problems with a cardinality constraint. Adv. Eng. Softw. **80**, 101–115 (2015)
24. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks No. IV, 27 Nov–1 Dec, pp. 1942–1948, Perth Australia (1995)
25. Kaveh, A.: Advances in Metaheuristic Algorithms for Optimal Design of Structures. Springer, New York (2014)
26. Yang, X.S.: Recent Advances in Swarm Intelligence and Evolutionary Computation (2015)
27. Gholizadeh, S., Fattahi, F.: Design optimization of tall steel buildings by a modified particle swarm algorithm. Struct. Des. Tall Spec. Build. **23**(4), 285–301 (2014)
28. Kaveh, A., Sheikholeslami, R., Talatahari, S., Keshvari-Ilkhichi, M.: Chaotic swarming of particles: a new method for size optimization of truss structures. Adv. Eng. Softw. **67**, 136–147 (2014)
29. Gandomi, A.H., Kashani, A.R., Mousavi, M., Jalalvandi, M.: Slope stability analyzing using recent swarm intelligence techniques. Int. J. Numer. Anal. Methods Geomech. (2014)
30. Montalvo, I., Izquierdo, J., Pérez-García, R., Herrera, M.: Water distribution system computer-aided design by agent swarm optimization. Comput. Civil Infrastruct. Eng. **29**(6), 433–448 (2014)
31. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**, 60–68 (2001)
32. Yoo, D.G., Kim, J.H., Geem, Z.W.: Overview of Harmony Search algorithm and its applications in civil engineering. Evol. Intel. **7**(1), 3–16 (2014)
33. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press, Bristol (2008)
34. Gandomi, A.H., Yang, X.S., Talatahari, S., Alavi, A.H.: Firefly algorithm with chaos. Commun. Nonlinear Sci. Numer. Simul. **18**(1), 89–98 (2013)
35. Yang, X.S.: Firefly algorithms for multimodal optimization. In: Stochastic Algorithms: Foundations and Applications, pp. 169–178. Springer, Heidelberg (2009)

36. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Mixed variable structural optimization using firefly algorithm. Comput. Struct. **89**(23), 2325–2336 (2011)
37. Talatahari, S., Gandomi, A.H., Yun, G.J.: Optimum design of tower structures using firefly algorithm. Struct. Des. Tall Spec. Build. **23**(5), 350–361 (2014)
38. Mauder, T., Sandera, C., Stetina, J., Seda, M.: Optimization of the quality of continuously cast steel slabs using the firefly algorithm. Mater. Technol. **45**(4), 347–350 (2011)
39. Miguel, L.F.F., Lopez, R.H., Miguel, L.F.F.: Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm. Adv. Eng. Softw. **56**, 23–37 (2013)
40. Liu, C., Gao, Z., Zhao, W.: A new path planning method based on firefly algorithm. In: 2012 Fifth International Joint Conference on Computational Sciences and Optimization (CSO), pp. 775–778 (2012)
41. Yang, X.S., Deb, S.: Cuckoo search via Lévy flights. In: World Congress on Nature and Biologically Inspired Computing, 2009. NaBIC 2009, pp. 210–214 (2009)
42. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng. Comput. **29**(1), 17–35 (2013)
43. Kaveh, A., Bakhshpoori, T.: Optimum design of steel frames using cuckoo search algorithm with Lévy flights. Struct. Des. Tall Spec. Build. **22**(13), 1023–1036 (2013)
44. Gandomi, A.H., Talatahari, S., Yang, X.S., Deb, S.: Design optimization of truss structures using cuckoo search algorithm. Struct. Des. Tall Spec. Build. **22**(17), 1330–1349 (2013)
45. Ouaarab, A., Ahiod, B., Yang, X.S.: Discrete cuckoo search algorithm for the travelling salesman problem. Neural Comput. Appl. **24**(7–8), 1659–1669 (2014)
46. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pp. 65–74. Springer, Berlin, Heidelberg (2010)
47. Yang, X.S., Hossein Gandomi, A.: Bat algorithm: a novel approach for global engineering optimization. Eng. Comput. **29**(5), 464–483 (2012)
48. Gandomi, A.H., Yang, X.S., Alavi, A.H., Talatahari, S.: Bat algorithm for constrained optimization tasks. Neural Comput. Appl. **22**(6), 1239–1255 (2013)
49. Gholizadeh, S., Shahrezaei, A.M.: Optimal placement of steel plate shear walls for steel frames by bat algorithm. Struct. Des. Tall Spec. Build. **24**(1), 1–18 (2015)
50. Kaveh, A., Zakian, P.: Enhanced bat algorithm for optimal design of skeletal structures. Asian J. Civil Eng. **15**(2), 179–212 (2014)
51. Talatahari, S., Kaveh, A.: Improved bat algorithm for optimum design of large-scale truss structures. Int. J. Optim. Civil Eng. **5**(2), 241–254 (2015)
52. Bekdas, G., Nigdeli, S. M., Yang, X.S.: Metaheuristic optimization for the design of reinforced concrete beams under flexure moments. In: Proceedings of the 5th European Conference of Civil Engineering (ECCIE'14), pp. 184–188 (2014)
53. Zhou, Y., Xie, J., Zheng, H.: A hybrid bat algorithm with path relinking for capacitated vehicle routing problem. Math. Probl. Eng. (2013)
54. Bozorg-Haddad, O., Karimirad, I., Seifollahi-Aghmiuni, S., Loáiciga, H.A.: Development and application of the bat algorithm for optimizing the operation of reservoir systems. J. Water Resour. Plann. Manag. (2014). doi:10.1061/(ASCE)WR.1943-5452.0000498
55. Erol, O.K., Eksin, I.: A new optimization method: Big Bang Big Crunch. Adv. Eng. Softw. **37**, 106–111 (2006)
56. Camp, C.V.: Design of space trusses using Big Bang-Big Crunch optimization. J. Struct. Eng. **133**(7), 999–1008 (2007)
57. Kaveh, A., Talatahari, S.: Size optimization of space trusses using Big Bang-Big Crunch algorithm. Comput. Struct. **87**(17), 1129–1140 (2009)
58. Kaveh, A., Talatahari, S.: A discrete Big Bang-Big Crunch algorithm for optimal design of skeletal structures. Asian J. Civil Eng. **11**(1), 103–122 (2010)
59. Hasançebi, O., Kazemzadeh Azad, S.: Discrete size optimization of steel trusses using a refined Big Bang–Big Crunch algorithm. Eng. Optim. **46**(1), 61–83 (2014)
60. Hasançebi, O., Azad, S.K.: An exponential Big Bang-Big Crunch algorithm for discrete design optimization of steel frames. Comput. Struct. **110**, 167–179 (2012)

61. Tang, H., Zhou, J., Xue, S., Xie, L.: Big Bang-Big Crunch optimization for parameter estimation in structural systems. Mech. Syst. Signal Process. **24**(8), 2888–2897 (2010)
62. Camp, C.V., Akin, A.: Design of retaining walls using Big Bang-Big Crunch optimization. J. Struct. Eng. **138**(3), 438–448 (2011)
63. Kaveh, A., Talatahari, A.: A novel heuristic optimization method: charged system search. Acta Mech. **213**, 267–289 (2010)
64. Kaveh, A., Maniat, M.:. Damage detection in skeletal structures based on charged system search optimization using incomplete modal data. Int. J. Civil Eng. **12**(2A), 292–299 (2014)
65. Kaveh, A., Shokohi, F.: Cost optimization of castellated beams using charged system search algorithm. Iran. J. Sci. Technol. Trans. Civil Eng. **38**(C1), 235–249 (2014)
66. Kaveh, A., Nasrollahi, A.: Charged system search and particle swarm optimization hybridized for optimal design of engineering structures. Sci. Iran. Trans. A Civil Eng. **21**(2), 295 (2014)
67. Kaveh, A., Massoudi, M.S.: Multi-objective optimization of structures using charged system search. Sci. Iran. Trans. A Civil Eng. **21**(6), 1845 (2014)
68. Kaveh, A., Mohammadi, S., Hosseini, O.K., Keyhani, A., Kalatjari, V.R.: Optimum parameters of tuned mass dampers for seismic applications using charged system search. Iran. J. Sci. Technol. Trans. Civil Eng. **39**(C1), 21 (2015)
69. Kaveh, A., Pirgholizadeh, S., Hosseini, O.K.: Semi-active tuned mass damper performance with optimized fuzzy controller using CSS algorithm. Asian J. Civil Eng. (BHRC) **16**(5), 587–606 (2015)
70. Gandomi, A.H., Alavi, A.H.: Krill Herd: a new bio-inspired optimization algorithm. Commun. Nonlinear Sci. Numer. Simul. **17**(12), 4381–4845 (2012)
71. Kaveh, A., Khayatazad, M.: A novel meta-heuristic method: ray optimization. Comput. Struct. **112–113**, 283–294 (2012)
72. Kaveh, A., Khayatazad, M.: Ray optimization for size and shape optimization of truss structures. Comput. Struct. **117**, 82–94 (2013)
73. Esmaeili, M., Zakeri, J.A., Kaveh, A., Bakhtiary, A., Khayatazad, M.: Designing granular layers for railway tracks using ray optimization algorithm. Sci. Iran. Trans. A Civil Eng. **22** (1), 47 (2015)
74. Yang, X. S. (2012), Flower pollination algorithm for global optimization. In: Unconventional Computation and Natural Computation 2012, Lecture Notes in Computer Science, vol. 7445, pp. 240–249
75. Den Hartog, J.P. (ed.): Mechanical Vibrations. Courier Corporation (1985)
76. Warburton, G.B., Ayorinde, E.O.: Optimum absorber parameters for simple systems. Earthq. Eng. Struct. Dyn. **8**(3), 197–217 (1980)
77. Warburton, G.B.: Optimum absorber parameters for various combinations of response and excitation parameters. Earthq. Eng. Struct. Dyn. **10**(3), 381–401 (1982)
78. Sadek, F., Mohraz, B., Taylor, A.W., Chung, R.M.: A method of estimating the parameters of tuned mass dampers for seismic applications. Earthq. Eng. Struct. Dyn. **26**(6), 617–636 (1997)
79. Rana, R., Soong, T.T.: Parametric study and simplified design of tuned mass dampers. Eng. Struct. **20**(3), 193–204 (1998)
80. Chang, C.C.: Mass dampers and their optimal designs for building vibration control. Eng. Struct. **21**(5), 454–463 (1999)
81. Lin, C.C., Wang, J.F., Ueng, J.M.: Vibration control identification of seismically excited MDOF structure-PTMD systems. J. Sound Vib. **240**(1), 87–115 (2001)
82. Aldemir, U.: Optimal control of structures with semiactive-tuned mass dampers. J. Sound Vib. **266**(4), 847–874 (2003)
83. Lee, C.L., Chen, Y.T., Chung, L.L., Wang, Y.P.: Optimal design theories and applications of tuned mass dampers. Eng. Struct. **28**(1), 43–53 (2006)
84. Bakre, S.V., Jangid, R.S.: Optimum parameters of tuned mass damper for damped main system. Struct. Control Health Monit. **14**(3), 448–470 (2007)

85. Hadi, M.N., Arfiadi, Y.: Optimum design of absorber for MDOF structures. J. Struct. Eng. **124**(11), 1272–1280 (1998)
86. Marano, G.C., Greco, R., Chiaia, B.: A comparison between different optimization criteria for tuned mass dampers design. J. Sound Vib. **329**(23), 4880–4890 (2010)
87. Singh, M.P., Singh, S., Moreschi, L.M.: Tuned mass dampers for response control of torsional buildings. Earthq. Eng. Struct. Dyn. **31**(4), 749–769 (2002)
88. Desu, N.B., Deb, S.K., Dutta, A.: Coupled tuned mass dampers for control of coupled vibrations in asymmetric buildings. Struct. Control Health Monit. **13**(5), 897–916 (2006)
89. Pourzeynali, S., Lavasani, H.H., Modarayi, A.H.: Active control of high rise building structures using fuzzy logic and genetic algorithms. Eng. Struct. **29**(3), 346–357 (2007)
90. Leung, A.Y.T., Zhang, H.: Particle swarm optimization of tuned mass dampers. Eng. Struct. **31**(3), 715–728 (2009)
91. Leung, A.Y., Zhang, H., Cheng, C.C., Lee, Y.Y.: Particle swarm optimization of TMD by non-stationary base excitation during earthquake. Earthq. Eng. Struct. Dynam. **37**(9), 1223–1246 (2008)
92. Steinbuch, R.: Bionic optimisation of the earthquake resistance of high buildings by tuned mass dampers. J. Bionic Eng. **8**(3), 335–344 (2011)
93. Bekdaş, G., Nigdeli, S.M.: Estimating optimum parameters of tuned mass dampers using harmony search. Eng. Struct. **33**(9), 2716–2723 (2011)
94. Bekdaş, G., Nigdeli, S.M.: Optimization of tuned mass damper with harmony search. In: Gandomi, A.H., Yang, X.-S., Alavi, A.H., Talatahari, S. (eds.) Metaheuristic Applications in Structures and Infrastructures, Chapter 14. Elsevier, Waltham (2013)
95. Bekdaş, G., Nigdeli, S.M.: Mass ratio factor for optimum tuned mass damper strategies. Int. J. Mech. Sci. **71**, 68–84 (2013)
96. Nigdeli, S.M., Bekdas, G.: Optimum tuned mass damper design for preventing brittle fracture of RC buildings. Smart Struct. Syst. **12**(2), 137–155 (2013)
97. Farshidianfar, A., Soheili, S.: Ant colony optimization of tuned mass dampers for earthquake oscillations of high-rise structures including soil–structure interaction. Soil Dyn. Earthq. Eng. **51**, 14–22 (2013)
98. Farshidianfar, A.: ABC optimization of TMD parameters for tall buildings with soil structure interaction. Interact. Multiscale Mech. **6**, 339–356 (2013)
99. Farshidianfar, A.: Optimization of TMD parameters for earthquake vibrations of tall buildings including soil structure interaction. Int. J. Optim. Civil Eng. **3**, 409–429 (2013)
100. Federal Emergency Management Agency (FEMA): Quantification of Building Seismic Performance Factors (2009)
101. Pacific Earthquake Engineering Research Center (PEER NGA DATABASE). http://peer.berkeley.edu/nga
102. Coello, C.C., Hernández, F.S., Farrera, F.A.: Optimal design of reinforced concrete beams using genetic algorithms. Expert Syst. Appl. **12**(1), 101–108 (1997)
103. Rafiq, M.Y., Southcombe, C.: Genetic algorithms in optimal design and detailing of reinforced concrete biaxial columns supported by a declarative approach for capacity checking. Comput. Struct. **69**(4), 443–457 (1998)
104. Camp, C.V., Pezeshk, S., Hansson, H.: Flexural design of reinforced concrete frames using a genetic algorithm. J. Struct. Eng. **129**(1), 105–115 (2003)
105. Govindaraj, V., Ramasamy, J.V.: Optimum detailed design of reinforced concrete frames using genetic algorithms. Eng. Optim. **39**(4), 471–494 (2007)
106. Fedghouche, F., Tiliouine, B.: Minimum cost design of reinforced concrete T-beams at ultimate loads using Eurocode2. Eng. Struct. **42**, 43–50 (2012)
107. Talatahari, S., Sheikholeslami, R., Shadfaran, M., Pourbaba, M.: Optimum design of gravity retaining walls using charged system search algorithm. Math. Probl. Eng. (2012)
108. Poursha, M., Khoshnoudian, F., Moghadam, A.S.: Harmony search based algorithms for the optimum cost design of reinforced concrete cantilever retaining walls. Int. J. Civil Eng. **9**(1), 1–8 (2011)

109. Bekdas, G., Nigdeli, S.M.: Optimization of T-shaped RC flexural members for different compressive strengths of concrete. Int. J. Mech. **7**, 109–119 (2013)
110. Lepš, M., Šejnoha, M.: New approach to optimization of reinforced concrete beams. Comput. Struct. **81**(18), 1957–1966 (2003)
111. Camp, C.V., Huq, F.: CO 2 and cost optimization of reinforced concrete frames using a Big Bang-Big Crunch algorithm. Eng. Struct. **48**, 363–372 (2013)
112. ACI 318 M-05: Building code requirements for structural concrete and commentary, American Concrete Institute, Farmington Hills, MI, USA (2005)

Application of the Flower Pollination Algorithm in Structural Engineering

Sinan Melih Nigdeli, Gebrail Bekdaş and Xin-She Yang

Abstract In the design of a structural system, the optimum values of design variables cannot be derived analytically. Structural engineering problems have various design constraints concerning structural security measures and practicability in production. Thus, optimization becomes an important part of the design process. Recent studies suggested that metaheuristic methods using random search procedures are effective for solving optimization problems in structural engineering. In this chapter, the flower pollination algorithm (FPA) is presented for dealing with structural engineering problems. The engineering problems are about pin-jointed plane frames, truss systems, deflection minimization of I-beams, tubular columns, and cantilever beams. The FPA inspired from the reproduction of flowers via pollination is effective to find the best optimum results when compared to other methods. In addition, the computing time is usually shorter and the optimum results are also robust.

Keywords Metaheuristic methods · Flower pollination algorithm · Structural optimization · Topology optimization · Weight optimization

S.M. Nigdeli · G. Bekdaş
Department of Civil Engineering, Istanbul University, 34320 Avcılar, Istanbul, Turkey
e-mail: melihnig@istanbul.edu.tr

G. Bekdaş
e-mail: bekdas@istanbul.edu.tr

X.-S. Yang (✉)
Design Engineering and Mathematics, Middlesex University London, The Burroughs, London NW4 4BT, UK
e-mail: x.yang@mdx.ac.uk; xy227@cam.ac.uk

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*, Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_2

1 Introduction

In solving optimization problems, traditional optimization methods such as gradient-based methods may not be able to cope with high nonlinearity and multimodality. Evolutionary algorithms and nature-inspired algorithms tend to produce better results for highly nonlinear problems. Such nature-inspired metaheuristic algorithms often imitate the successful nature of some biological, physical, or chemical systems in nature. They often have several processes as numerical, algorithmic steps in solving an optimization problem. Each metaheuristic algorithm can have different inspiration from the nature and special rules according to the process of the natural systems. Detailed information about several metaheuristic algorithms can be found in the literature [1, 2]. Inspiration and pioneer papers of several metaheuristic algorithms are given in Table 1.

In structural engineering, economy is one of the main goals of the design engineering. The optimum design variables ensuring security measures and the minimum cost cannot be found with linear equations. As the equations and system behavior can be highly nonlinear, iterative numerical algorithms have been employed to find a solution. Using metaheuristic algorithms, the global optimum solution can be found more effectively.

In this chapter, the flower pollination algorithm (FPA) developed by Yang [16] is presented. Several structural optimization problems were investigated using FPA and the optimum results were compared with other optimization methods.

Table 1 Metaheuristic algorithms and inspirations

Algorithm	Inspiration
Genetic algorithm [3, 4]	Darwinian evolution in nature
Simulated annealing [5]	Annealing process of materials
Ant colony optimization [6]	Behavior of ants foraging
Bee algorithm [7]	Behavior of bees
Particle swarm optimization [8]	Swarming behavior of birds and fish
Tabu search [9]	Human memory
Harmony search [10]	Musical performance
Big bang big crunch [11]	Evolution of the universe
Firefly algorithm [1]	Flashing characteristic of fireflies
Cuckoo search [12]	Brood parasitic behavior of cuckoo species
Charged system search [13]	Electrostatic and Newtonian mechanic laws
Bat algorithm [14]	Echolocation characteristic of microbats
Eagle strategy [15]	Foraging behavior of eagles
Flower pollination [16]	Pollination of flowering plants
Ray optimization [17]	Refraction of light

2 Flower Pollination Algorithm

In nature, the main purpose of the flowers is reproduction via pollination. Flower pollination is related to the transfer of pollen, which is done by pollinators such as insects, birds, bats, other animals or wind. Some flower types have special pollinators for successful pollination. The four rules of pollination have been formulated based on the inspiration from flowering plants and they form the main updating equations of the flower pollination algorithm [16].

1. Cross-pollination occurs from the pollen of a flower of different plants. Pollinators obey the rules of a Lévy distribution by jumping or flying distant steps. This is known as global pollination process.
2. Self-pollination occurs from the pollen of the same flower or other flowers of the same plant. It is local pollination.
3. Flower constancy is the association of pollinators and flower types. It is an enhancement of the flower pollination process.
4. Local pollination and global pollination are controlled by a probability between 0 and 1, and this probability is called as the switch probability.

In the real world, a plant has multiple flowers and the flower patches release a lot of pollen gametes. For simplicity, it is assumed that each plant has one flower producing a single pollen gamete. Due to this simplicity, a solution (x_i) in the present optimization problem is equal to a flower or a pollen gamete. For multi-objective optimization problems, multiple pollen gametes can be considered.

In the flower pollination algorithm, there are two key steps involving global and local pollination. In the global pollination step, the first and third rules are used together to find the solution of the next step (x_i^{t+1}) using the values from the previous step (step t) defined as x_i^t. Global pollination is formulized in Eq. (1).

$$x_i^{t+1} = x_i^t + L\left(x_i^t - g*\right) \tag{1}$$

The subscript i represents the ith pollen (or flower) and Eq. (1) is applied for the pollen of the flowers. g^* is the current best solution. L is the strength of the pollination, which is drawn from a Lévy distribution.

The second rule is used for local pollination with the third rule about flower constancy. The new solution is generated with random walks as seen in Eq. (2).

$$x_i^{t+1} = x_i^t + \varepsilon\left(x_j^t - x_k^t\right) \tag{2}$$

where x_j^t and x_k^t are solutions of different plants. ε is randomized between 0 and 1. According to the fourth rule, a switch probability (p) is used in order to choose the type of pollination which will control the optimization process in iterations.

The details of the optimization process can be seen in the pseudocode which is given for the flower pollination algorithm.

Objective minimize or maximize f(x), x=(x_1, x_2,…. x_d)
Initialize a population of n flowers or pollen gametes with random numbers
Find the best solution (g_*) *of the initial population*
Define a switch probability (*p*)
while (t<Number of iterations)
for *i*=1:*n* (*n is the number of flowers or pollen in the population*)
if *rand*<*p*
Global pollination using Eq.(1)
else
Local pollination using Eq.(2)
end if
Evaluate new solutions
Update the better solutions in the population
end for
Find the current best solution (g_*)
end while

Flower pollination algorithm was first proposed for the optimization problems with a single objective. Then, Yang et al. developed a multi-objective approach for FPA [18].

3 Numerical Examples

In this chapter, six numerical examples are investigated using the FPA. They are pin-jointed plane frame optimization, truss system optimization, vertical deflection minimization of an I-beam, cost optimization of a tubular column, and weight optimization of cantilever beams (two types of cantilever beams).

3.1 Pin-Jointed Plane Frame Optimization Problem

A pin-jointed plane frame with five members is given in Fig. 1. The system is symmetrical and thus only half of the system with three members is optimized for the minimum weight. Topology optimization is done to find the optimum values of θ_1 and θ_2 angles shown in the figure. The system has a fixed base. This example was originally given by Majid [19]. The vertical deflections of joints 1 and 2, used in the design constraints, are defined as

$$|\Delta_1(\theta_1, \theta_2)| \leq \text{Max } \Delta, \tag{3}$$

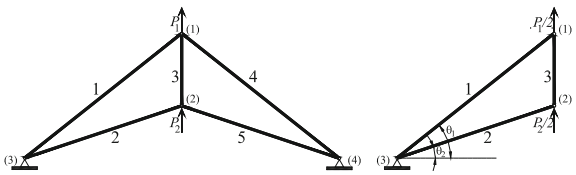

Fig. 1 The optimized system [19]

and

$$|\Delta_2(\theta_1, \theta_2)| \leq \text{Max } \Delta, \tag{4}$$

where θ_1 and θ_2 are searched within a range defined as minimum ($\theta_{\min}$) and maximum ($\theta_{\max}$) limits.

The members have a constant cross-sectional area (A) and an elasticity modulus (E). P_1 and P_2 loads are used in the system. The length between the supports is defined as l.

The lengths of the members are defined in Eqs. (5)–(7) for members 1, 2, and 3, respectively.

$$l_1 = \frac{l}{2\cos(\theta_1)} \tag{5}$$

$$l_2 = \frac{l}{2\cos(\theta_2)} \tag{6}$$

$$l_3 = \frac{l}{2\cos(\theta_1)\cos(\theta_2)} \sqrt{\cos^2(\theta_1) + \cos^2(\theta_2) - 2\cos(\theta_1)\cos(\theta_2)2\cos(\theta_1 - \theta_2)} \tag{7}$$

Since the members of the system have the same cross-sectional area, the total length of the system can be taken as the optimization objective in order to minimize the overall weight. The objective function is shown in Eq. (8).

$$\text{Minimize } f(\theta_1, \theta_1) = \sum_{i=1}^{3} l_i \tag{8}$$

If $\Delta = (\Delta_1, \Delta_1)^t$, $K\Delta = F$ where K is the stiffness matrix and F is the load vector. Since K is equal to $B^T kB$, these matrices are given in Eqs. (9) and (10).

$$k = \begin{bmatrix} \frac{EA}{l_1} & 0 & 0 \\ 0 & \frac{EA}{l_2} & 0 \\ 0 & 0 & \frac{EA}{l_3} \end{bmatrix} \tag{9}$$

$$B = \begin{bmatrix} \sin(\theta_1) & 0 \\ 0 & \sin(\theta_2) \\ 1 & -1 \end{bmatrix} \tag{10}$$

Thus, the stability equation ($K\Delta = F$) of the system can be written as

$$EA \begin{bmatrix} \frac{\sin^2(\theta_1)}{l_1} + \frac{1}{2l_3} & -\frac{1}{2l_3} \\ -\frac{1}{2l_3} & \frac{\sin^2(\theta_2)}{l_2} + \frac{1}{2l_3} \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_1 \end{bmatrix} = \begin{bmatrix} \frac{P_1}{2} \\ \frac{P_2}{2} \end{bmatrix} \tag{11}$$

and Δ is then solved in the optimization process starting from random θ_1 and θ_2 values.

The optimization process has been carried out for the design constants given in Table 2. The results of the flower pollination algorithm were compared with the other methods employing GA [20] and CS [12].

The optimum results together with the results of other approaches are given in Table 3. The table shows that the proposed method is effective to find better solutions.

The numerical example is done by taking the switch probability as 0.5 and the number of population as 5. The optimum solution is found at the 1609th iteration. The convergence of the optimization is seen in the objective function versus iterations as shown in the plot given in Fig. 2.

As seen in the optimum results, θ_1 is nearly equal to θ_2. Since the objective function is the minimization of the total length, the length of the third member is nearly zero. Thus, the method is effective to find the global optimum value.

Also, the optimization process is done for different cross-sectional areas and force values. In all these cases, P_2 is taken as half of P_1. The optimum results of the

Table 2 Design constants of numerical example

Max Δ	5 mm
θ_{min}	0
θ_{max}	$\pi/3$ (rad)
A	100 mm^2
E	200,000 MPa
P_1	100 kN
P_2	50 kN
L	1000 mm

Table 3 Optimum results of the numerical example

Method	θ_1 (rad)	θ_2 (rad)	F (θ_1, θ_2)
FPA	0.477634	0.477133	1125.87
GA	0.475784	0.472764	1125.98
CS	0.477459	0.477446	1125.92

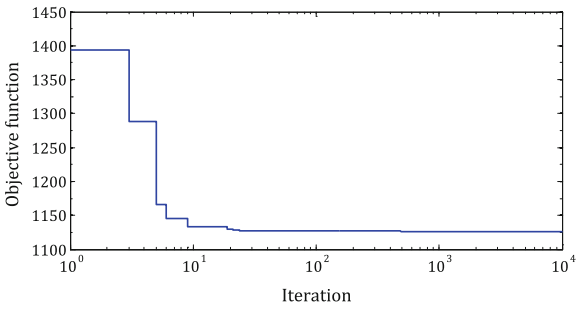

Fig. 2 Objective function versus iteration

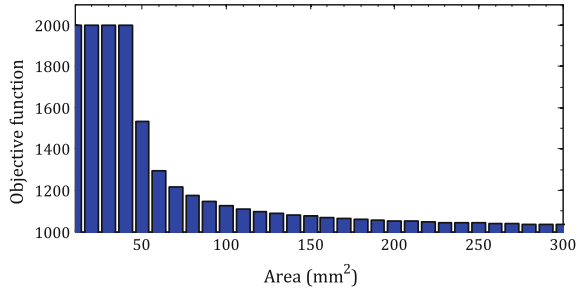

Fig. 3 Optimum results for different cross sections

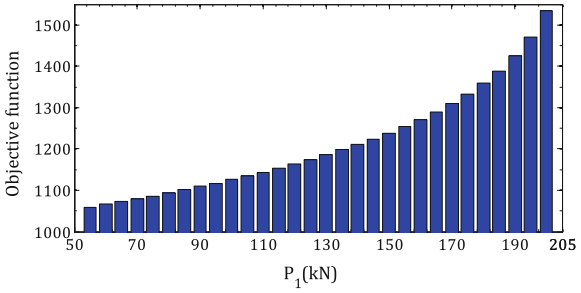

Fig. 4 Optimum results for different forces

objective function for different cross-sectional areas and forces are given in Figs. 3 and 4, respectively. In Fig. 3, the upper limits of the angles were found as the optima for small cross-sectional areas.

3.2 A Three-Bar Truss System Optimization Problem

A three-bar truss structure is given in Fig. 5. This problem was first presented in Nowcki [21]. The objective function is about the minimization of the volume of the truss structure and this function is given in Eq. (12).

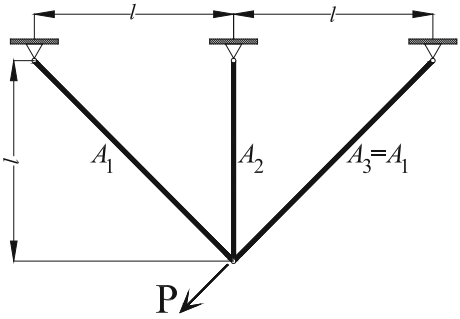

Fig. 5 Truss optimization problem

$$\text{Minimize}: f(A_1, A_2) = \left(2\sqrt{2}A_1 + A_2\right)l \tag{12}$$

The design variables are the cross-sectional areas of structural members. Since the system is symmetric, only cross sections shown with A_1 and A_2 are optimized. The optimization problem is carried out for stress constraints. These constraints are

$$g_1 = \frac{\sqrt{2}A_1 + A_2}{\sqrt{2}A_1^2 + 2A_1A_2}P - \sigma \leq 0, \tag{13}$$

$$g_2 = \frac{A_2}{\sqrt{2}A_1^2 + 2A_1A_2}P - \sigma \leq 0, \tag{14}$$

$$g_3 = \frac{1}{A_1 + \sqrt{2}A_2}P - \sigma \leq 0. \tag{15}$$

The cross-sectional areas were searched for the ranges; $0 \leq A_1 \leq 1$ and $0 \leq A_2 \leq 1$. The length, the load, and the stress limit were taken as l = 100 cm; P = 2 kN, and σ = 2 kN/cm^2, respectively. The optimum results are summarized in Table 4 together with the optimum results obtained by other optimization methods.

The result of Tsai [24] seems to be lower than the present results, but the result of Tsai [24] is not acceptable because one of the design constraints (defined by g_1) is slightly violated in their study. Using the values of the design variables, the stress on a bar does not obey the stress constraint and the security of the system is not provided.

Table 4 Optimization results

st_max	Park et al. [22]	Ray and Saini [23]	Tsai [24]	Yang and Gandomi [12]	Gandomi et al. [14]	Present study
A_1	0.78879	0.79500	0.788	0.78863	0.78867	0.78853
A_2	0.40794	0.39500	0.408	0.40838	0.40902	0.40866
f_{min}	263.8965	264.3000	263.68	263.8962	263.9716	263.8958

3.3 Vertical Deflection Minimization Problem of an I-Beam

The FPA is also tested for the problem presented by Gold and Krishnamurty [25]. The optimization objective is to minimize the vertical deflection of an I-beam as shown in Fig. 6.

The vertical deflection of an I-beam is depended to design load (P), length of the beam (L), and modulus of elasticity which are taken as 600 kN, 200 cm, and 20000 kN/cm^2, respectively. The load (Q) in the other direction is taken as 50 kN.

The deflection of a beam is defined by

$$f(x) = \frac{PL^3}{48EI} \tag{16}$$

and the objective function of numerical example can be written as Eq. (17) when the design constants and the moment of inertia (I) of the I-beam are defined in the Eq. (16).

$$\text{Minimize } f(b,h,t_w,t_f) = \frac{5000}{\frac{t_w(h-2t_f)^3}{12} + \frac{bt_f^3}{6} + 2bt_f\left(\frac{h-t_f}{2}\right)^2} \tag{17}$$

According to the objective function given in Eq. (17), the design variables are h, b, t_w, and t_f. The ranges of these variables are

$$\begin{aligned} 10 \le h \le 80, \\ 10 \le b \le 50, \\ 0.9 \le t_w \le 5 \text{ and} \\ 0.9 \le t_f \le 5. \end{aligned} \tag{18}$$

The cross section of an I-beam must be <300 cm^2 and the allowable bending stress of the beam is 6 kN/cm^2. In that case, the cross section and stress constraints can be written as Eqs. (19) and (20):

$$g_1 = 2bt_w + t_w(h - 2t_f) \le 300, \tag{19}$$

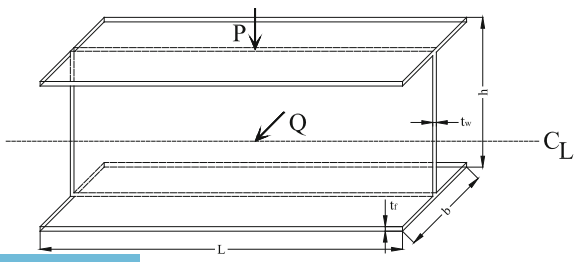

Fig. 6 I-beam problem

Table 5 Optimum results for I-beam example

	CS	ARSM	Improved ARSM	Present study
h	80.00	80.00	79.99	80.00
b	50.00	37.05	48.42	50.00
t_w	0.9	1.71	0.9	0.9
t_f	2.3216715	2.31	2.40	2.3217922
F_{min}	0.0130747	0.0157	0.131	0.0130741

$$g_1 = \frac{180000h}{t_w(h-2t_f)^3 + 2bt_w(4t_f^2 + 3h(h-2t_f))} + \frac{15000b}{t_w^3(h-2t_f) + 2t_w b^3} \leq 6. \quad (20)$$

The optimum result obtained by the flower pollination algorithm was compared with the results by other methods such as the adaptive response surface method (ARSM), improved ARSM [26], and cuckoo search [12]. The results are presented in Table 5.

The optimum value of FPA has been obtained for 25 pollen agents and 5000 evaluations of design variables. With the increase in iterations, the algorithm stops if further improvement for the optimum results cannot be obtained. In the actual runs of the algorithm, the only improvement is the difference between the worst and best results in this case.

Comparing to the results of CS, a minor improvement of the optimum results can be seen, but in the engineering design, it is not very important. In addition to best optimum results, the convergence and minimization of computational time are also important for metaheuristic algorithms. Also, the same optimum results must be obtained for various runs of the optimization process. For a population of 25 pollens and for a fixed number of 5000 evaluations, the same results were obtained for every run. For 3000 evaluations, the best optimum results are generally found. Even to increase the number of evaluations to 15000, the maximum and minimum objective functions are generally the same. This shows the stability and robustness of the algorithm.

3.4 Cost Optimization of Tubular Column Under Compressive Load

The tubular column is shown in Fig. 7 [27]. The tubular column is axially loaded with a load (P), and the upper and the lower bounds of the columns are supported from hinged bearings. The design constants of the optimization are shown in Table 6.

The types of constraint about compressive and buckling are important for the column. The compressive stress of the column must be lower than the yield stress of the tubular column. This constraint is given as Eq. (21).

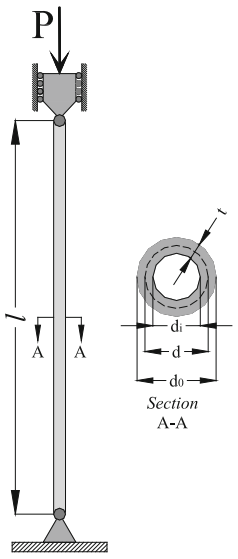

Fig. 7 Tubular column and *A-A* cross-section

Table 6 Design constants of the tubular column

Symbol	Definition	Value
P	Axial force	2500 kgf
σ_y	Yield stress	500 kgf/cm^2
E	Modulus of elasticity	0.85x10^6 kgf/cm^2
ρ	Density	0.0025 kgf/cm^3
L	Length of column	250 cm

$$g_1 = \frac{P}{\pi dt\sigma_y} - 1 \leq 0 \tag{21}$$

The axial load must be lower than the buckling load of the column defined as the Euler buckling load:

$$P_{kr} = \frac{\pi^2 EI}{l^2} \tag{22}$$

where I is the moment of inertia of the tubular column section. When Eq. (22) is modified for the column section, g_2; the constraint is formed as given in Eq. (23):

$$g_2 = \frac{8PL^2}{\pi^3 Edt(d^2 + t^2)} - 1 \leq 0. \tag{23}$$

The objective function is to minimize

$$f(d,t) = 9.8dt + 2d, \tag{24}$$

which is the sum of the material and construction costs of the tubular column.

The ranges of the design variables found in objective function can also be given as constraints. In this example, the diameter (d) of the column must be between 2 and 14 cm, while the thickness (t) of the column is a variable between 0.2 and 0.8 cm. These ranges are formulized as the following constraints:

$$g_3 = \frac{2.0}{d} - 1 \leq 0 \tag{25}$$

$$g_4 = \frac{d}{14} - 1 \leq 0 \tag{26}$$

$$g_5 = \frac{0.2}{t} - 1 \leq 0 \tag{27}$$

$$g_6 = \frac{t}{0.8} - 1 \leq 0 \tag{28}$$

The optimization is done for 25 pollens and 200 iterations. The total running time of the optimization algorithm is <0.1 s. The statistical results of the optimization results are shown in Table 7. Nearly the best and the worst results are equal to each other.

The results are compared with the results by other methods and are summarized in Table 8. The FPA-based approach is more effective than the other methods. In addition, the convergence of the algorithm is very effective.

Table 7 Statistical results of optimization of tubular column example

No. pollen	No. evals.	Best	Average	Worst	St. deviation
25	5000	26.4994969	26.499497	26.4994974	1.699×10^{-7}

Table 8 Optimum results for tubular column example

	Hsu and Liu [27]	Rao [28]	CS [12]	Present study
d	5.4507	5.44	5.45139	5.451160
t	0.292	0.293	0.29196	0.291965
g_1	-3.45×10^{-5}	−0.8579	0.0241	9.4343×10^{-7}
g_2	1.32×10^{-4}	0.0026	−0.1095	-4.249×10^{-7}
g_3	−0.6331	−0.8571	−0.6331	−0.6331
g_4	−0.6107	0	−0.6106	−0.6106
g_5	−0.3151	−0.7500	−0.3150	−0.3150
g_6	−0.6350	0	−0.6351	−0.6350
F_{min}	26.4991	26.5323	26.53217	26.49948

3.5 *Weight Optimization of Cantilever Beams*

Two types of cantilever beams are optimized using FPA. In the first example, (Fig. 8) a beam with square section is investigated. Also, the inner part of the section is empty. The second example (Fig. 9) beam has a rectangular cross section.

3.5.1 Weight Optimization of Cantilever Beams (Example 1)

The example given by Fleury and Braibant [29] is optimized by using FPA. The cantilever beam is shown in Fig. 8. The beam is rigidly supported from one end and the other end is free. A vertical load is applied from the free end of the beam. The cantilever beam is optimized for an objective function

$$\text{Minimize } f(X) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5) \tag{29}$$

subject to

$$g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0. \tag{30}$$

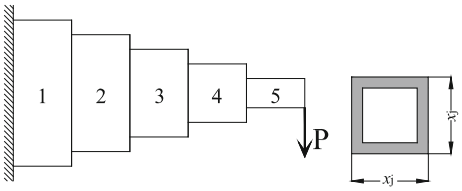

Fig. 8 The cantilever beam (Example 1)

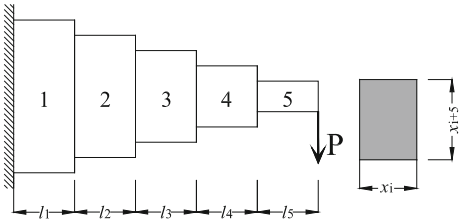

Fig. 9 The cantilever beam (Example 2)

The beam is divided into five steps with different cross sections. The thickness (t) is taken as 2/3, and it is fixed for each step of the cantilever beam. For all design variables from 1 to 5 (j = 1− 5), the following ranges

$$0.01 \leq x_j \leq 100 \tag{31}$$

are also taken into consideration.

The optimum results were compared with the results by CS [12] and other methods [30] as shown in Table 9.

The objective function is the same for all methods because the sensitivity of the results of the other methods is not known. Comparing to CS, it is possible to find the optimum results with 25 pollens and 300 search iterations while CS is performed for 50 cuckoos and 2500 search iterations.

3.5.2 Cantilever Beam Optimization (Example 2)

The cantilever beam shown in Fig. 9 contains ten design variables. This example was originally given by Thanedar and Vanderplaats [31]. The cross section of the beam is rectangular. The first five design variables are the width (x_1–x_5) of the cantilever beam. The height of the beam (x_6–x_{10}) is the other variable. The optimization objective is given below.

$$\text{Minimize} \quad V = \sum_{i=1}^{5} x_i x_{i+5} l_i \tag{32}$$

The length of a step (l_i) is fixed and 100 cm in value. The optimization is done considering 11 constraints formulized as

$$g_1 = \frac{600P}{x_5 x_{10}^2} - 14{,}000 \leq 0 \tag{33}$$

Table 9 The optimum results of cantilever beam (Example 1)

Methods	x_1	x_2	x_3	x_4	x_5	F_{min}
MMA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA(I)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA(II)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
Present study	6.0202	5.3082	4.5042	3.4856	2.1557	1.33997

CONLIN CONvex LINearization, *MMA* method of moving asymptotes, *GCA* generalized convex approximation

$$g_2 = \frac{6P(l_s + l_4)}{x_4 x_9^2} - 14{,}000 \leq 0 \tag{34}$$

$$g_3 = \frac{6P(l_s + l_4 + l_3)}{x_3 x_8^2} - 14{,}000 \leq 0 \tag{35}$$

$$g_4 = \frac{6P(l_s + l_4 + l_3 + l_2)}{x_2 x_7^2} - 14{,}000 \leq 0 \tag{36}$$

$$g_5 = \frac{6P(l_s + l_4 + l_3 + l_2 + l_1)}{x_1 x_6^2} - 14{,}000 \leq 0 \tag{37}$$

$$g_6 = \frac{Pl^3}{3E}\left(\frac{1}{I_s} + \frac{7}{I_4} + \frac{19}{I_3} + \frac{37}{I_2} + \frac{61}{I_1}\right) - 2.7 \leq 0 \tag{38}$$

$$g_7 = \frac{x_{10}}{x_5} - 20 \leq 0 \tag{39}$$

$$g_8 = \frac{x_9}{x_4} - 20 \leq 0 \tag{40}$$

$$g_9 = \frac{x_8}{x_3} - 20 \leq 0 \tag{41}$$

$$g_{10} = \frac{x_7}{x_2} - 20 \leq 0 \tag{42}$$

$$g_{11} = \frac{x_6}{x_1} - 20 \leq 0 \tag{43}$$

The solution range ares

$$1 \leq x_i \leq 5 \text{ for } i = 1 \text{ to } 5 \tag{44}$$

and

$$30 \leq x_i \leq 65 \text{ for } i = 1 \text{ to } 5 \tag{45}$$

In addition, P = 50,000 kN and E = 2×10^7 N/cm^2. The optimum results are presented in Table 10. The FPA algorithm is effective to find the minimum objective function value of the numerical example.

Table 10 The optimum results of cantilever beam (Example 2)

	Thanedar and Vanderplaats [31]	Lamberti and Pappalettere [32]	Huang and Arora [33]	BA [14]	Present study
x_1	3.06	–	–	2.99204	2.98211
x_2	2.81	–	–	2.77756	2.77002
x_3	2.52	–	–	2.52359	2.51546
x_4	2.2	–	–	2.20455	2.19861
x_5	1.75	–	–	1.74977	1.74722
x_6	61.16	–	–	59.84087	59.94777
x_7	56.24	–	–	55.55126	55.6512
x_8	50.47	–	–	50.4718	50.58328
x_9	44.09	–	–	44.09106	44.17907
x_{10}	35.03	–	–	34.99537	35.02744
Best objective	63110	65352.2	63108.7	61914.9	61849.9

4 Conclusion

From the extensive discussions in this chapter, it can be concluded that the FPA is an effective and suitable algorithm for solving structural engineering problems. It is also easy to implement.

For the pin-jointed plane frame optimization problem, the best optimum results have been obtained by FPA comparing to GA [20] and CS [12]. In addition, the problem has been solved for different loads. As the load increases, the objective function (total length of bars) also increases.

The optimization results of three-bar truss system by the FPA have also been compared with the results by CS [12], bat algorithm [14], and several other methods [22–24]. The results of FPA are slightly better than the results of other methods without exceeding the design constraints.

For the third example, the vertical deflection of an I-beam has been minimized. An important reduction of the existing best results is not provided, but similar results were obtained with a slight improvement using FPA. The major advantage is the shorter computation time and the robustness of the method because the optimum results were obtained for a much lower number of function evaluations than that in CS [13].

The tubular column design under a compressive load is a major structural engineering problem. The results for this example have been compared with the results by CS [12] and several other approaches. The results obtained by FPA are more effective than others.

The last examples are about two types of cantilever beams and the weight optimization of structural elements has been carried out. Comparing with other

methods, the improvement of the result is not physically meaningful for the first cantilever beam example; however, the results are obtained after a much lower number of iterations comparing with those in CS [12]. For the second cantilever beam example with 10 variables and 11 constraints, FPA is very effective and has obtained much better results.

All the above confirm that FPA is a feasible algorithm for optimization in structural engineering by providing better designs with less computing time and improving the robustness of finding the best optimum values. The effectiveness of FPA can be attributed to the fact that it is a good combination of local search (self-pollination) and global search (cross-pollinations). It can be expected that FPA can be used to solve many other optimization problems.

References

1. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2008)
2. Yang, X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications. Wiley, New York (2010)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Boston (1989)
4. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
5. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
6. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B **26**, 29–41 (1996)
7. Nakrani, S., Tovey, C.: On honey bees and dynamic allocation in an internet server colony. Adapt. Behav. **12**(3–4), 223–240 (2004)
8. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks No. IV, 27 Nov–1 Dec, pp. 1942–1948, Perth Australia (1995)
9. Glover, F.: Heuristic for integer programming using surrogate constraints. Decis. Sci. **8**, 156–166 (1977)
10. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**, 60–68 (2001)
11. Erol, O.K., Eksin, I.: A new optimization method: big bang big crunch. Adv. Eng. Softw. **37**, 106–111 (2006)
12. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng. Comput. **29**, 17–35 (2013)
13. Kaveh, A., Talatahari, A.: A novel heuristic optimization method: charged system search. Acta Mech. **213**, 267–289 (2010)
14. Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. Eng. Comput. **29**(5), 464–483 (2012)
15. Yang, X.S., Deb, S.: Two-stage eagle strategy with differential evolution. Int. J. Bio-Inspired Comput. **4**(1), 1–5 (2012)
16. Yang, X.S.: Flower pollination algorithm for global optimization. In: Unconventional Computation and Natural Computation 2012. Lecture Notes in Computer Science, vol. 7445, pp. 240–249 (2012)

17. Kaveh, A., Khayatazad, M.: A novel meta-heuristic method: ray optimization. Comput. Struct. **112–113**, 283–294 (2012)
18. Yang, X.S., Karamanoglu, M., He, X.: Flower pollination algorithm: a novel approach for multiobjective optimization. Eng. Optim. **46**(9), 1222–1237 (2012)
19. Majid, K.I.: Optimum design of structures. Newnes-Butterworth, London (1974)
20. Li, J.P., Balazs, M.E., Parks, G.T.: Engineering design optimization using species-conserving genetic algorithms. Eng. Optm. **39**(2), 147–161 (2007)
21. Nowcki, H.: Optimization in pre-contract ship design. In: Fujita, Y., Lind, K., Williams, T. J. (eds.) Computer Applications in the Automation of Shipyard Operation and Ship Design, vol. 2, pp. 327–338. Elsevier, New York (1974)
22. Park, Y.C., Chang, M.H., Lee, T.Y.: A new deterministic global optimization method for general twice differentiable constrained nonlinear programming problems. Eng. Optim. **39**(4), 397–411 (2007)
23. Ray, T., Saini, P.: Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng. Optm. **33**(6), 735–748 (2001)
24. Tsai, J.: Global optimization of nonlinear fractional programming problems in engineering design. Eng. Optim. **37**(4), 399–409 (2005)
25. Gold, S., Krishnamurty, S.: Trade-offs in robust engineering design. In: Proceedings of the 1997 ASME Design Engineering Technical Conferences, DETC97/DAC3757, 14–17 Sept, Saramento, California (1997)
26. Wang, G.G.: Adaptive response surface method using inherited latin hypercube design points. Trans. ASME **125**, 210–220 (2003)
27. Hsu, Y.L., Liu, T.C.: Developing a fuzzy proportionalderivative controller optimization engine for engineering design optimization problems. Eng. Optm. **39**(6), 679–700 (2007)
28. Rao, S.S.: Engineering optimization: theory and practice, 3rd edn. Wiley, Chichester (1996)
29. Fleury, C., Braibant, V.: Structural optimization: a new dual method using mixed variables. Int. J. Numer. Meth. Eng. **23**, 409–428 (1986)
30. Chickermane, H., Gea, H.C.: Structural optimization using a new local approximation method. Int. J. Numer. Meth. Eng. **39**, 829–846 (1996)
31. Thanedar, P.B., Vanderplaats, G.N.: Survey of discrete variable optimization for structural design. J. Struct. Eng. ASCE **121**(2), 301–306 (1995)
32. Lamberti, L., Pappalettere, C.: Move limits definition in structural optimization with sequential linear programming. Part II Numer. Ex. Comput. Struct. **81**, 215–238 (2003)
33. Huang, M.W., Arora, J.S.: Optimal design with discrete variables: some numerical experiments. Int. J. Numer. Meth. Eng. **40**, 165–188 (1997)

Use of Swarm Intelligence in Structural Steel Design Optimization

Mehmet Polat Saka, Serdar Carbas, Ibrahim Aydogdu and Alper Akin

Abstract In this chapter, the optimum design problem of steel space frames is formulated according to the provisions of LRFD-AISC (Load and Resistance Factor Design-American Institute of Steel Corporation). The weight of the steel frame is taken as the objective function to be minimized. The design optimization problem necessitates selection of steel sections for the members of the steel frame from the available steel profiles lists. This turns the design optimization problem into discrete programming problem. Obtaining the optimum solution of such programming problems is cumbersome with mathematical programming techniques. On the other hand with the use of recently developed metaheuristic techniques that are based on swarm intelligence, the solution of the same problem becomes straightforward. Five different structural optimization algorithms are developed which are based on ant colony optimization, particle swarm optimizer, artificial bee colony algorithm, firefly algorithm, and cuckoo search algorithm, respectively. Two real size steel space frames; one rigidly connected and the other pin jointed are designed using each of these algorithms. The optimum designs obtained by these techniques are compared and performance of each version is evaluated. It is noticed that most of swarm intelligence-based algorithms are simple and robust techniques that determine the optimum solution of structural design optimization problems efficiently without requiring much of a mathematical struggle.

M.P. Saka (✉)
Department of Civil Engineering, University of Bahrain, Isa Town, Bahrain
e-mail: mpsaka@uob.edu.bh

S. Carbas
Department of Civil Engineering, Karamanoglu Mehmetbey University, Karaman, Turkey
e-mail: scarbas@kmu.edu.tr

I. Aydogdu
Department of Civil Engineering, Akdeniz University, Antalya, Turkey
e-mail: aydogdu@akdeniz.edu.tr

A. Akin
Thomas & Betts Corporation, Meyer Steel Structures, Memphis, TN 38125, USA
e-mail: alperakin@yahoo.com

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*,
Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_3

Keywords Structural design optimization · Load and resistance factor design (LRFD) · Swarm intelligence · Ant colony algorithm · Particle swam optimizer · Artificial bee colony algorithm · Firefly algorithm · Cuckoo search algorithm

1 Introduction

Competitiveness of today's economy and diminishing of world's limited resources are forcing structural designers to come up with structures that require just sufficient material to be built, while its response to external loads is within design code limitations. Undoubtedly, excessive use of materials in structural design not only yields expensive structures that cause constructors to loose tenders but also consumes more of natural sources and adds more pollution to the atmosphere which triggers global warming. Hence, it is apparent that in order to have a sustainable development, structures are required to be designed and built using sufficient amount of material but not more. Structural design optimization tools exactly try to achieve this goal. They aim to design steel structures such that the steel frames have the minimum weight (minimum material) and in the meantime the response of the frame under the external loads that the frame may be subjected to during its lifetime is within the design code limitations. Design of steel structures has its own features and not similar to the design of other structures. Designer cannot use any section she/he may desire but to select among the set of steel profiles available in practice for beams and columns of the frame under consideration. This selection is required to be carried out in such a way that the frame with the selected steel profiles should have the displacements less than those prescribed in the design code and its members have sufficient strength to satisfy the design strength limitations under the external loads as described in steel design code provisions. In the meantime, it is the desire of structural designer that the cost of the structure is the minimum.

In this chapter, first the design optimization problem of steel space frames according to the provisions of LRFD-AISC (Load and Resistance factor Design—American Institution of Steel Corporation) [1] is presented. The aim of the optimum design is to minimize the weight of the steel frame which becomes the objective function in the programming problem. The objective function and the design constraints as given by LRFD-AISC are described in detail in the following sections. It should be noticed that structural steel design in practice requires selection of steel profiles from the available steel profile list for the members of the steel frame under consideration for design. Therefore, the structural steel design optimization problem requires finding out the steel profile designations from the available set of steel sections for the members of a steel frame such that with these steel sections the weight of the steel frame is the minimum and its behavior under the external loads within the limitations described in steel design code.

The mathematical model of such optimization problem turns out to be a nonlinear discrete programming problem. Among the mathematical programming techniques available to obtain the solution of such discrete variables, problem is sequential linear discrete programming algorithm [2]. This technique attempts to obtain the solution of nonlinear discrete programming problem by solving series of mixed-integer linear programming problems. The nonlinear problems are linearized about an initially selected point using first-order Taylor's series expansion. However, because of the discrete variables the mixed-integer linear programming technique cannot be applied directly. The discrete variables are to be redefined by means of another set of design variables where the new design variables are to be equal either zero or one in the solution. These additional design variables increase the total number of variables in the problem needlessly which makes the coding of the algorithm quite cumbersome. The success of obtaining the optimum solution is closely related with the quality of the selection of the initial design point. Furthermore, computational difficulties such as convergence problems can be stumbled in large size optimization problems.

The metaheuristic techniques that are emerged in last three decades do not suffer the above discrepancies [3–10]. They are nontraditional stochastic search and optimization methods that quite effective in finding the optimum solution of combinatorial optimization problems similar to the one described in the preceding section. They do not require the gradient information of the objective function and constraints and they use probabilistic transition rules not deterministic rules. These methods move within a design domain randomly with the aim of reaching the optimum solution. However, this random move is not based on a blind way of searching for the optimum in a confined design region but it makes use of an intelligent heuristics to guide the search. This is why stochastic search methods are also called metaheuristic algorithms. The fundamental properties of metaheuristic algorithms are that they imitate certain strategies taken from nature, social culture, biology, or laws of physics that direct the search process. The strategies employed in search of optimum solution in these techniques simulate the natural phenomena. Among the metaheuristic algorithms, those who are based on swarm intelligence are shown to be robust, effective, and quite powerful techniques in obtaining near optimum solutions if not optimum in engineering design optimization problems [10]. There are several reviews available in the literature which comprehensively summarizes the use of metaheuristic algorithms in the design optimization of steel skeletal structures [11–16].

In this chapter, five different structural design optimization algorithms are developed which are based on ant colony optimization, particle swarm optimizer, artificial bee colony algorithm, firefly algorithm, and cuckoo search algorithm, respectively. Two real size steel space frames are designed using each of these algorithms considering the LRFD-AISC provisions. The optimum designs obtained by these techniques are compared and performance of each version is evaluated.

2 Discrete Optimum Design of Steel Space Frames to LRFD-AISC

The design of steel space frames necessitates the selection of steel sections for its columns and beams from standard steel section tables such that the frame satisfies the serviceability and strength requirements specified by the code of practice while the economy is observed in the overall or material cost of the frame. When the design constraints are implemented from LRFD-AISC, the following nonlinear discrete programming problem is obtained.

2.1 The Objective Function

The objective function is taken as the minimum weight of the frame which is expressed as follows:

$$\text{Minimize} \quad W = \sum_{r=1}^{\text{ng}} m_r \sum_{s=1}^{t_r} \ell_s \tag{1}$$

where W defines the weight of the frame, m_r is the unit weight of the steel section selected from the standard steel sections table that is to be adopted for group r, t_r is the total number of members in group r, ng is the total number of groups in the frame, and l_s is the length of members which belongs to group r.

2.2 Strength Constraints

For the case where the effect of warping is not included in the computation of the strength capacity of W sections that are selected for beam–column members of the frame, the following inequalities given in Chapter H of LRFD-AISC are required to be satisfied:

$$\text{for } \frac{P_u}{\phi P_n} \geq 0.2;\ g_{s,i} = \frac{P_u}{\phi P_n} + \frac{8}{9}\left(\frac{M_u}{\phi_b M_{nx}} + \frac{M_u}{\phi_b M_{ny}}\right) \leq 1,0 \tag{2}$$

$$\text{for } \frac{P_u}{\varphi P_n} < 0.2;\ g_{s,i} = \frac{P_u}{2\phi\, P_n} + \left(\frac{M_u}{\phi_b M_{nx}} + \frac{M_u}{\phi_b M_{ny}}\right) \leq 1,0 \tag{3}$$

where M_{nx} is the nominal flexural strength at strong axis (x axis), M_{ny} is the nominal flexural strength at weak axis (y axis), M_{ux} is the required flexural strength at strong axis (x axis), M_{uy} is the required flexural strength at weak axis (y axis), P_n is the nominal axial strength (tension or compression), and P_u is the required axial strength (tension or compression) for member i and ℓ represents the loading case. The values of M_{ux} and M_{uy} are required to be obtained by carrying out $P-\Delta$ analysis of the steel frame. This is an iterative process which is quite time consuming. In Chapter C of LRFD-AISC, an alternative procedure is suggested for the computations of M_{ux} and M_{uy} values. In this procedure, two first-order elastic analyses are carried out. In the first, frame is analyzed under the gravity loads only where the sway of the frame is prevented to obtain M_{nt} values. In the second, the frame is analyzed only under the lateral loads to find M_{lt} values. These moment values are then combined using the following equation as given in the design code:

$$M_u = B_1 M_{nt} + B_2 M_{lt} \tag{4}$$

where B_1 is the moment magnifier coefficient and B_2 is the sway moment magnifier coefficient. The details of how these coefficients are calculated are given in Chapter C of LRFD-AISC [1]. Equations (2) and (3) represent strength constraints for doubly and singly symmetric steel members subjected to axial force and bending. If the axial force in member k is tensile force, the terms in these equations are given as follows: P_{uk} is the required axial tensile strength, P_{nk} is the nominal tensile strength, ϕ becomes ϕ_t in the case of tension and called strength reduction factor which is given as 0.90 for yielding in the gross section and 0.75 for fracture in the net section, ϕ_b is the strength reduction factor for flexure given as 0.90, M_{uxk} and M_{uyk} are the required flexural strength, and M_{nxk} and M_{nyk} are the nominal flexural strengths about major and minor axis of member k, respectively. It should be pointed out that required flexural bending moment should include second-order effects. LRFD suggests an approximate procedure for computation of such effects which is explained in C1 of LRFD. In the case the axial force in member k is compressive force, the terms in Eqs. (2) and (3) are defined as follows: P_{uk} is the required compressive strength, P_{nk} is the nominal compressive strength, and ϕ becomes ϕ_c which is the resistance factor for compression given as 0.85. The remaining notations in Eqs. (2) and (3) are the same as the definition given above.

The nominal tensile strength of member k for yielding in the gross section is computed as $P_{nk} = F_y A_{gk}$ where F_y is the specified yield stress and A_{gk} is the gross area of member k. The nominal compressive strength of member k is computed as $P_{nk} = A_{gk} F_{cr}$ where $F_{cr} = \left(0.658^{\lambda_c^2}\right) F_y$ for $\lambda_c \leq 1.5$ and $F_{cr} = \left(0.877/\lambda_c^2\right) F_y$ for $\lambda_c > 1.5$ and $\lambda_c = \frac{Kl}{r\pi}\sqrt{F_y/E}$. In these expressions, E is the modulus of elasticity, and K and l are the effective length factor and the laterally unbraced length of member k, respectively.

2.3 Displacement Constraints

The lateral displacements and deflection of beams in steel frames are limited by the steel design codes due to serviceability requirements. According to the ASCE Ad Hoc Committee report [17], the accepted range of drift limits in the first-order analysis is 1/750–1/250 times the building height H with a recommended value of H/400. The typical limits on the inter-story drift are 1/500–1/200 times the story height. Based on this report, the deflection limits recommended are proposed in [18–20] for general use which is repeated in Table 1.

2.3.1 Deflection Constraints

It is necessary to limit the mid-span deflections of beams in a steel space frame not to cause cracks in brittle finishes that they may support due to excessive displacements. Deflection constraints can be expressed as an inequality limitation as shown in the following:

$$g_{dj} = \frac{\delta_{jl}}{\delta_j^u} - 1 \leq 0 \quad j = 1, \ldots, n_{\text{sm}}, \; l = 1, \ldots, n_{\text{lc}} \tag{5}$$

where δ_{jl} is the maximum deflection of jth member under the lth load case, δ_j^u is the upper bound on this deflection which is defined in the code as span/360 for beams carrying brittle finishers, n_{sm} is the total number of members where deflections limitations are to be imposed, and n_{lc} is the number of load cases.

2.3.2 Drift Constraints

These constraints are of two types. One is the restriction applied to the top story sway and the other is the limitation applied on the inter-story drift.

Top Story Drift Constraint
Top story drift limitation can be expressed as an inequality constraint as shown in the following:

Table 1 Displacement limitations for steel frames

	Item	Deflection limit
1	Floor girder deflection for service live load	L/360
2	Roof girder deflection	L/240
3	Lateral drift for service wind load	H/400
4	Inter-story drift for service wind load	H/300

$$g_{tdj} = \frac{(\Delta_{\text{top}})_{jl}}{H/\text{Ratio}} - 1 \leq 0 \quad j = 1, \ldots, n_{\text{jtop}}, \; l = 1, \ldots, n_k \tag{6}$$

where H is the height of the frame, n_{jtop} is the number of joints on the top story, n_{lc} is the number of load cases, and $(\Delta_{\text{top}})_{jl}$ is the top story drift of the jth joint under lth load case.

Inter-Story Drift

In multi-story steel frames, the relative lateral displacements of each floor are required to be limited. This limit is defined as the maximum inter-story drift which is specified as h_{sx}/Ratio where h_{sx} is the story height and ratio is a constant value given in ASCE Ad Hoc Committee report [14].

$$g_{idj} = \frac{(\Delta_{oh})_{jl}}{\text{h}_{\text{sx}}/\text{Ratio}} - 1 \leq 0 \quad j = 1, \ldots, n_{\text{st}}, \; l = 1, \ldots, n_{\text{lc}} \tag{7}$$

where n_{st} is the number of story, n_{lc} is the number of load cases, and $(\Delta_{oh})_{jl}$ is the story drift of the jth story under lth load case.

2.4 Geometric Constraints

In steel frames, it is desired that column section for upper floor should not have a larger section than the lower story column for practical reasons, because having a larger section for upper floor requires a special joint arrangement which is neither preferred nor economical. The same applies to the beam-to-column connections. The W section selected for any beam should have a flange width smaller than or equal to the flange width of the W section selected for the column to which the beam is to be connected. These are shown in Fig. 1 and named as geometric constraints. These limitations are included in the design optimization model to satisfy practical requirements. Two types of geometric constraints are considered in the mathematical model. These are column-to-column geometric constraints and beam-to-column geometric limitations.

2.4.1 Column-to-Column Geometric Constraints

The depth and the unit weight of W sections selected for the columns of two consecutive stores should be either equal to each other or the one in the upper story should be smaller than the one in the lower story. These limitations are included in the design problem as inequality constraints as shown in the following:

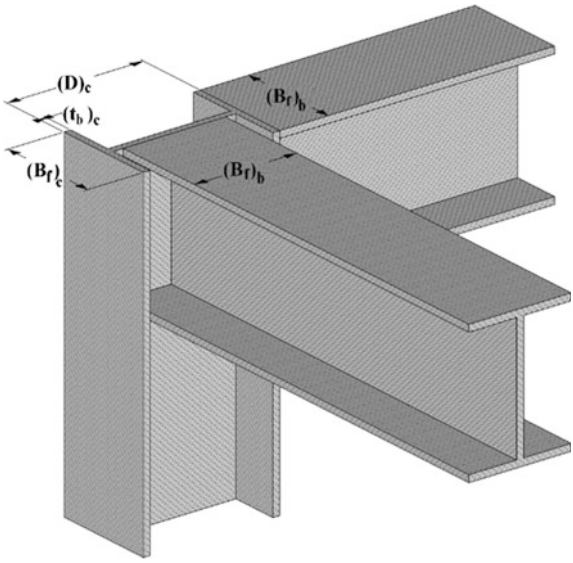

Fig. 1 Beam column geometric constraints

$$g_{cdi} = \frac{D_i}{D_{i-1}} - 1 \leq 0 \quad i = 2...n_j \tag{8}$$

$$g_{cmi} = \frac{m_i}{m_{i-1}} - 1 \leq 0 \quad i = 2...n_j \tag{9}$$

where n_j is the number of stories, m_i is the unit weight of W section selected for column story i, m_{i-1} is the unit weight of W section selected for of column story $(i-1)$, D_i is the depth of W section selected for of column story i, and D_{i-1} is the depth of W section selected for of column story $(i-1)$.

2.4.2 Beam-to-Column Geometric Constraints

When a beam is connected to a flange of a column, the flange width of the beam should be less than or equal to the flange width of the column so that the connection can be made without difficulty. In order to achieve this, the flange width of the beam should be less than or equal to $(D - 2t_b)$ of the column web dimensions in the connection where D and t_b are the depth and the flange thickness of W section, respectively, as shown in Fig. 1:

$$g_{bci} = \frac{(B_f)_{bi}}{D_{ci} - 2(t_{bc})_i} - 1 \leq 0 \quad i = 1...n_{j1} \tag{10}$$

or

$$g_{bbi} = \frac{(B_f)_{bi}}{(B_f)_{ci}} - 1 \leq 0 \quad i = 1 \ldots n_{j2} \tag{11}$$

where n_{j1} is the total number of joints where beams are connected to the web of a column, n_{j2} is the total number of joints where beams connected to the flange of a column, D_{ci} is the depth of W section selected for the column at joint i,$(t_b)_{cj}$ is the flange thickness of W section selected for the column at joint i, $(B_f)_{ci}$ is the flange width of W section selected for the column at joint i, and $(B_f)_{bi}$ is the flange width of W section selected for the beam at joint i.

The optimum design of steel space frames problem described in preceding sections where the objective function is given in Eq. (1) and the constraints are depicted from Eq. (2) to (11) is a nonlinear discrete optimization problem. It is apparent that in order to determine the optimum solution of this problem steel designer has to find out the suitable combination of W sections that makes the frame weight minimum and in the same time the design code provisions are all satisfied. Here the selection of a W section from an available steel profile list is carried out by choosing an integer number from a set which consists of integer numbers starting from 1 to the total number of sections in the list. This integer number is the sequence number of that particular W section. Hence, the design solution is a set of integer numbers each of which represents the sequence number of W section in the design pool. This is a combinatorial optimization problem.

3 Swarm Intelligence-Based Metaheuristic Optimization Algorithms

Metaheuristic optimization algorithms differ from that of mathematical programming techniques in the fact that they do not employ gradient descent or quasi-Newton techniques but heuristic search. A heuristic exploits problem-depended information to find a sufficiently good solution to a specific problem, while metaheuristic is a general-purpose algorithm that can be applied almost any type of optimization problem. In reality, metaheuristic is also heuristic, but a more powerful one that can be viewed as upper level general method with a guiding strategy in designing underlying heuristic. The rule of thumb selected for exploration of the optimum solution by metaheuristic algorithms may be borrowed from nature or social culture. In general, a metaheuristic is an iterative process with set of concepts that are used for exploring and exploiting the search space to determine the best solution among the alternative solutions. Metaheuristic algorithms are not problem specific, approximate, and usually non-deterministic. It is important that there should be a dynamic balance between diversification and intensification in metaheuristic procedure. Diversification generally refers to the

exploration of the search space and intensification refers to the exploitation of the accumulated search experience. The balance between these two concepts is important so as not to waste too much time in regions of the search space which does not possess high-quality solutions while the algorithm can quickly find out the regions of high-quality solutions [3–10].

Some of the recent metaheuristic techniques are based on swarm intelligence. A swarm is a large number of homogenous, unsophisticated agents that interact locally among themselves, and their environment, without any central control or management to yield a global behavior to emerge. Biologists are amazed with for example what an ant or bee colony achieves. Although single ant or a bee is not smart individual, their colonies are smart. When it comes to deciding what to do next, most ants or bees do not have any clue. However, as a swarm, they can find the shortest route to the best food source, allocate workers to different tasks, and defend a territory from invaders without having someone in control or as manager. As an individual they are tiny dummies, but they respond quickly and effectively to their environment as colonies. This is what is called swarm intelligence [10]. It is the collective behavior of decentralized and self-organized natural or artificial systems. Metaheuristic optimization techniques based on swarm intelligence are made up population of simple agents interacting with one another and with their environment as is the case in ant colonies, bird flocking, animal herding, and fish schooling. These techniques imitate the behavior of these colonies in a numerical optimization procedure. They employ what for example an ant colony uses to find the shortest route between their nest and food source as a guiding mechanism in order to search design domain to find the optimum solution of an optimization problem. They also utilize some additional strategies to avoid getting trapped in confined areas of search domain. Latest applications have shown that they are robust, effective, and quite powerful in obtaining near optimum solutions if not optimum in engineering design optimization problems [3, 10].

Performance evaluation of different metaheuristic algorithms in structural steel design optimization is carried out in the literature [21, 22]. This chapter intends to examine the use of swarm intelligence-based structural design optimization algorithms. For this purpose, five structural design optimization algorithms are developed each of which based on ant colony algorithm, particle swarm optimizer, artificial bee colony algorithm, firefly algorithm, and cuckoo search algorithm. Ant colony algorithm is inspired from the way that ant colonies find the shortest route between the food source and their nest [23–25]. Particle swarm optimizer mimics the social behavior of bird flocking [26, 27]. Artificial bee colony algorithm imitates the foraging behavior of honey bee colony [28, 29]. Firefly algorithm imitates the idealized behavior of flashing characteristics of fireflies. These insects communicate, search for pray, and find mates using bioluminescence with varying flaying patterns [30–32]. Cuckoo search algorithm simulates reproduction strategy of cuckoo birds [33].

Some of the optimization problems do not have any constraints and they only require minimizing or maximizing the objective function. However, most of the optimization problems in practice do have constraints. Swarm intelligence-based algorithms can only handle unconstrained optimization problems. Hence, it

becomes necessary to transform the optimization problem with constraints into the one which is unconstrained. This is achieved by either using a penalty function. There are several penalty function methods. Very comprehensive review of these techniques is covered [34]. One of the commonly used penalty functions is given in the following:

$$W_p = W(1 + C)^{\varepsilon} \tag{12}$$

where W is the value of objective function of optimum design problem given in Eq. (1), W_p is the penalized weight of structure, C is the value of total constraint violations which is calculated by summing the violation of each individual constraint, and ε is penalty coefficient which is taken as 2.0 in this work as suggested in [21, 22].

$$c = \sum_{i=1}^{\mathrm{nc}} c_i \tag{13}$$

$$c_i = \begin{cases} 0 & \text{if} \quad g_j \leq 0 \\ g_j & \text{if} \quad g_j > 0 \end{cases} \qquad j = 1, \ldots, \mathrm{nc} \tag{14}$$

where g_j is the jth constraint function and nc is the total number of constraints in the optimum design problem. Constraint functions for the steel frame are given through in Eqs. (2)–(11). It should be reminded that all the constraints are required to be normalized before they are used in the metaheuristic algorithms. A solution vector which may be slightly infeasible in one or more constraints is also included among the feasible solutions. This is called adaptive error strategy which is explained in [35].

3.1 Ant Colony Optimization (ACO)

Ant colony optimization technique is inspired from the way that ant colonies find the shortest route between the food source and their nest. Ants being completely blind individuals can successfully discover as a colony the shortest path between their nest and the food source. They manage this through their typical characteristic of employing volatile substance called pheromones. They perceive these substances through very sensitive receivers located in their antennas. The ants deposit pheromones on the ground when they travel which is used as a trail by other ants in the colony. When there is choice of selection for an ant between two paths, it selects the one where the concentration of pheromone is more. Since the shorter trail will be reinforced more than the long one after a while a great majority of ants in the colony will travel on this route. Ant colony optimization algorithm is first used in finding the solution of traveling salesman [23–25]. The steps of ant colony optimization are as follows:

1. Select number of ants each of which represents a potential solution to the optimization problem. Define a pool for each decision variable in the optimization problem which consists of possible values that particular variable can take. Assign randomly each ant to each decision variable. Calculate initial pheromone amount (τ_0) as $\tau_0 = 1/Z_{\min}$ where $Z_{\min}$ is the minimum value of the objective function without penalty violation. The pheromone and visibility values for each decision variable are calculated defined as given below:

$$\tau_{ij} = \frac{1}{\tau_0}, \quad v_{ij} = \frac{1}{x_{ij}}, \quad i = 1, 2, \ldots, \text{nv}\ j = 1, 2, \ldots, \text{npool} \tag{15}$$

where x_{ij} is jth value of the design variable i, nv is the total number of design variables in the optimization problem, and npool is the total number of values in the pool from which a value can be selected for decision variable i.

2. Each ant in the colony selects its first design variables. Ants then select values for their decision variables from the value pool. This selection is carried out by a decision process which depends on probability computation given below:

$$P_{ij}(t) = \frac{\left(\tau_{ij}(t)\right)^{\alpha} \cdot \left(v_{ij}\right)^{\beta}}{\sum_{j \in \text{allowed}}^{\text{ndiv}} \left(\tau_{ij}(t)\right)^{\alpha} \cdot \left(v_{ij}\right)^{\beta}} \quad i = 1, 2, \ldots, \text{nv}\ j = 1, 2, \ldots, \text{npool} \tag{16}$$

where $P_{ij}(t)$ is the probability of jth value selected from the pool for decision variable i which is assigned to ant at time t. α and β are the parameters which are used to arrange the influence of local trail values and visibility, respectively. This process continues until all ants assign values for their first design variables.

3. Apply local update rule at the end of the each tour. Concentration of pheromone for values selected by ants from the pool is lowered in order to promote exploration in the search. This corresponds to the evaporation of the pheromone substance in the real life. The mathematical expression for this is $\tau_{ij}(t) = \zeta \cdot \tau_{ij}(t)$ where ζ is called local update coefficient whose value changes from 0 to 1. If the value of this parameter is selected close to 1, fast convergence occurs and the algorithm may end up with a local optimum. On the other hand, if the value is chosen close to 0 convergence difficulties arise in the problem.
4. Start assigning a value from the pool for the next decision variable. Continue this assigning procedure until all the ants in the colony have a value for each decision variable. At the end of the tour, apply local update rule. This procedure continues until all ants assign values for all decision variables. At the end of this process, each ant has a selected value for each decision variable and all together each ant represents a candidate solution for the optimization problem.

5. Apply global update scheme using Eq. (17):

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{17}$$

where ρ is constant selected between 0 and 1. $(1-\rho)$ represents the evaporation amount of pheromone between time t and $t+n$ (the amount of time required to complete a cycle). $\Delta\tau_{ij}$ is the change in pheromone amount on the path connecting decision variable i to decision variable j. Value of $\Delta\tau_{ij}$ is represented by the following formula:

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{18}$$

where k represents any ant from 1 to m (where m is the number of ants) and $\Delta\tau_{ij}^{k}$ is the change in the pheromone for ant k. Calculation of $\Delta\tau_{ij}^{k}$ term is described as follows:

$$\Delta\tau_{ij}^{k} = \frac{1}{Z_k} \tag{19}$$

where Z_k is the penalized objective function value for ant k. This is the end of one ant colony algorithm cycle. To start another cycle, all ants are returned to their initial decision variables and above steps are replicated again.

6. Repeat steps 2–5 until the termination criterion is satisfied which generally taken as the maximum number of iterations.

3.2 *Particle Swarm Optimizer (PSO)*

Particle swarm optimizer is based on the social behavior of animals such as fish schooling, insect swarming, and birds flocking [26, 27]. The method considers an artificial swarm which consists of particles (agents). The behavior of each agent in the swarm is simulated according to three rules. The first is *separation* where each agent tries to move away from its neighbors if they are too close. The second is *alignment* where each agent steers toward the average heading of its neighbors. The third is *cohesion* where each agent tries to go toward the average position of its neighbors. This simulation is extended to have roost by [27]. They have amended the above three rules as follows: each agent is attracted toward the location of roost; each agent remembers where it was closer to the roost and each agent shared information with all other agents about its closest location to the roost.

The particle swarm optimizer selects a number of particles to represent a swarm. Each particle in the swarm is a potential solution to the optimization problem under consideration. A particle explores the search domain by moving around. This move is decided by making use of its own experience and the collective experience of the

swarm. Each particle has three main parameters: position, velocity, and fitness. Position represents the decision variables of the optimization problem, velocity determines the rate of change of the position, and fitness is the value of the objective function at the particle's current position. The fitness value is a measure of how good is the solution it represents for the optimization problem. The algorithm starts solving an optimization problem by first initializing each particle. In the initiation phase, each particle is given a random initial position and an initial velocity. The position of a particle represents a solution of the problem and has therefore a value, given by the objective function. While moving in the search space, particles memorize the position of the best solution they found. At each iteration of the algorithm, each particle moves with a velocity that is a weighted sum of three components: the old velocity, a velocity component that drives the particle toward the location in the search space where it previously found the best solution so far, and a velocity component that drives the particle towards the location in the search space where the neighbor particles found the best solution so far. There are several variants of particle swarm algorithm. The steps of the standard algorithm are outlined in the following:

1. Initialize swarm of particles with positions x_0^i and initial velocities v_0^i randomly distributed throughout the design space. These are obtained from the following expressions:

$$x_0^i = x_{\min} + r(x_{\max} - x_{\min}) \tag{20}$$

$$v_0^i = [(x_{\min} + r(x_{\max} - x_{\min}))/\Delta t] \tag{21}$$

   where the term r represents a random number between 0 and 1, and $x_{\min}$ and $x_{\max}$ represent the design variables upper and lower bounds, respectively.
2. Evaluate the objective function values $f(x_k^i)$ using the design space positions x_k^i.
3. Update the optimum particle position p_k^i at the current iteration k and the global optimum particle position p_k^g.
4. Update the position of each particle from $x_{k+1}^i = x_k^i + v_{k+1}^i \Delta t$ where x_{k+1}^i is the position of particle i at iteration $k + 1$, v_{k+1}^i is the corresponding velocity vector, and Δt is the time step value.
5. Update the velocity vector of each particle. There are several formulas for this depending on the particular particle swarm optimizer under consideration.

$$v_{k+1}^i = w v_k^i + c_1 r_1 \frac{(p_k^i - x_k^i)}{\Delta t} + c_2 r_2 \frac{(p_k^g - x_k^i)}{\Delta t} \tag{22}$$

   where r_1 and r_2 are the random numbers between 0 and 1, p_k^i is the best position found by particle i so far, and p_k^g is the best position in the swarm at time k. w is the inertia of the particle which controls the exploration properties of the algorithm. c_1 and c_2 are the trust parameters that indicate how much confidence the particle has in itself and in the swarm, respectively. Usually, they are taken as 2.

6. Repeat steps 2–5 until the stopping criterion is met.

In order to control the change of particles' velocities, upper and lower bounds for the velocity change are also limited to a user-specified value of $v_{\max}$. Once the new position of a particle is calculated, using Eq. (20), the particle then flies toward it. The main parameters used in the particle swarm optimizer are the population size (number of particles), number of generation cycles, and the maximal change of a particle velocity of $v_{\max}$.

3.3 *Artificial Bee Colony Algorithm (ABC)*

Artificial bee colony algorithm mimics the intelligent foraging behavior of a honey bee colony [28]. In the artificial bee colony algorithm, there are three types of bees which carry out different tasks. The first group of bees is the *employed bees* that locate food source, evaluate its amount of nectar, and keep the location of better sources in their memory. These bees when fly back to hive share this information to other bees in the dancing area by dancing. The dancing time represents the amount of nectar in the food source. The second group is the *onlooker bees* who observe the dance and may decide to fly to the food source if they find it is worthwhile to visit the food source. Therefore, food sources reach in the amount of nectar attract more onlooker bees. The third group is the *scout bees* that explore new food sources in the vicinity of the hive randomly. The employed bee whose food source has been abandoned by the bees becomes a scout bee. Overall, scout bees carry out the exploration, employed, and onlooker bees perform the task of exploitation. Each food source is considered as a possible solution for the optimization problem and the nectar amount of a food source represents the quality of the solution which is identified by its fitness value. The artificial bee colony algorithm consists of four stages. These stages are initialization phase, employed bees phase, onlooker bees phase, and scout bees phase.

1. ***Initialization phase***: Initialize all the vectors of the population of food sources, $x_p, p = 1, \ldots, np$ using Eq. (23) where np is the population size (total number of artificial bees). Each food source is a solution vector consisting of n variables $(x_{pi}, i = 1, \ldots, n)$ and it is a potential solution to the optimization problem under consideration.

$$x_{pi} = x_{li} + \text{rand}(0, 1)(x_{ui} - x_{li}) \tag{23}$$

   where $x_{\ell i}$ and x_{ui} are the upper and lower bounds on x_i.
2. ***Employed bees phase***: Employed bees search new food sources using Eq. (24):

$$v_{pi} = x_{pi} + f_{pi}(x_{pi} - x_{ki}) \tag{24}$$

   where $k \neq i$ is a randomly selected food source, and φ_{pi} is a random number in range $[-1, 1]$. After producing the new food source, its fitness is calculated. If its

fitness is better than x_{pi} the new food source replaces the previous one. The fitness value of the food sources is calculated according to Eq. (25):

$$\text{fitness}(x_p) = \begin{cases} \frac{1}{1+f(x_p)} & \text{if} \quad f(x_p) \geq 0 \\ 1 + \text{abs}(f(x_p)) & \text{if} \quad f(x_p) < 0 \end{cases} \tag{25}$$

3. ***Onlooker bees phase***: Unemployed bees consist of two groups. These are onlooker bees and scouts. Employed bees share their food source information with onlooker bees. Onlooker bees choose their food source depending on the probability value P_p which is calculated using the fitness values of each food source in the population as shown in Eq. (26):

$$P_p = \frac{\text{fitness}(x_p)}{\sum_{p=1}^{np} \text{fitness}(x_p)} \tag{26}$$

After a food source x_{pi} for an onlooker bee is probabilistically chosen, a neighborhood source is determined using Eq. (25) and its fitness value is computed using Eq. (26).

4. ***Scout bees phase***: The unemployed bees who choose their food sources randomly called scouts. Employed bees whose solutions cannot be improved after predetermined number of trials become scouts and their solutions are abandoned. These scouts start to search for new solutions.
5. Phases 2–4 are repeated until the termination criterion is satisfied.

3.4 *Firefly Algorithm (FFA)*

Firefly algorithm is based on the idealized behavior of flashing characteristics of fireflies [30, 31]. These insects communicate, search for pray, and find mates using bioluminescence with varying flaying patterns. The firefly algorithm is based on three rules, and they are as follows:

1. All fireflies are unisex so they attract one another.
2. Attractiveness is propositional to firefly brightness. For any couple of flashing fireflies, the less bright one moves toward the brighter one. Attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
3. The brightness of a firefly is affected or determined by the landscape of the objective function.

Mathematical interpretation of the above rules is given in the following as explained in [30].

Attractiveness: In the firefly algorithm, attractiveness of a firefly is assumed to be determined by its brightness which is related with the objective function. The

brightness i of a firefly at a particular location x can be chosen as $I(x) \propto f(x)$ where $f(x)$ is the objective function. However, the attractiveness β is relative, and it should be judged by the other fireflies. Thus, it will vary with the distance r_{ij} between firefly i and firefly j. In the firefly algorithm, the attractiveness function is taken to be proportional to the light intensity by adjacent fireflies and it is defined as follows:

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \quad (m \geq 1) \tag{27}$$

where β_0 is the attractiveness at $r = 0$.

Distance: The distance between any two fireflies i and j at x_i and x_j is calculated as

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k}\right)^2} \tag{28}$$

where $x_{i,k}$ is the kth component of the spatial coordinate x_i of the ith firefly.

Movement: The movement of a firefly i which is attracted to another brighter firefly j is determined by

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha\left(\text{rand} - \frac{1}{2}\right) \tag{29}$$

where the second term is due to the attraction and the third term is the randomization with α being the randomization parameter. "*rand*" is a random number generator uniformly distributed in [0, 1].

The parameter α_t essentially controls the randomness and to some extent, the diversity of solutions where t is the iteration number. This parameter can be tuned during iterations so that it can vary with the iteration counter t. The expression of $\alpha_t = \alpha_0\delta^t$ $(0 < \delta < 1)$ is suggested in [31] where α_0 is the initial randomness scaling factor, and δ is essentially a cooling factor. It is also stated in [31] that for most applications, the values from 0.95 to 0.97 is found suitable for δ. Regarding the initial α_0, simulations show that firefly algorithm is more efficient if α_0 is associated with the scaling of design variables. Let L be the average scale of the problem of interest, and it is set as $\alpha_0 = 0.01L$ initially. The factor 0.01 comes from the fact that random walk requires a number of steps to reach the target while balancing the local exploitation without jumping too far in a few steps [31]. The parameter β controls the attractiveness, and parametric studies suggest that $\beta_0 = 1$ can be used for most applications. However, it is mentioned in [31] that γ should be also related to the scaling L as $\gamma = 1/\sqrt{L}$.

The steps of the firefly algorithm are given below:

1. Generate initial population of n fireflies $\{x_i\}, (i = 1, 2, \ldots, n)$ randomly each of which represents a candidate solution to the optimization problem with objective function of $f(x)$ and decision variables $\{x\} = \{x_1, x_2, \ldots, x_m\}^T$.

2. Compute light intensity using Eq. (27) for each firefly $\{\beta\} = \{\beta_1, \beta_2, \ldots, \beta_n\}^T$. The distance between fireflies is computed from Eq. (28).
3. Move each firefly i toward other brighter fireflies using Eq. (29). If there is other brighter firefly move it randomly.
4. Evaluate new solutions and update light intensity.
5. Rank the fireflies and find the current best solution.
6. Repeat steps 2–5 until the termination criterion is satisfied.

3.5 Cuckoo Search Algorithm (CSA)

Cuckoo search algorithm simulates reproduction strategy of cuckoo birds [33]. Some species of cuckoo birds lay their eggs in the nests of other birds so that when the eggs are hatched their chicks are fed by the other birds. Sometimes, they even remove existing eggs of host nest in order to give more probability of hatching of their own eggs. Some species of cuckoo birds are even specialized to mimic the pattern and color of the eggs of host birds so that host bird could not recognize their eggs which give more possibility of hatching. In spite of all these efforts to conceal their eggs from the attention of host birds, there is a possibility that host bird may discover that the eggs are not its own. In such cases, the host bird either throws these alien eggs away or simply abandons its nest and builds a new nest somewhere else. In cuckoo search algorithm, cuckoo egg represents a potential solution to the design problem which has a fitness value. The algorithm uses three idealized rules as given in [33]. These are as follows: (a) each cuckoo lays one egg at a time and dumps it in a randomly selected nest. (b) The best nest with high-quality eggs will be carried over to the next generation. (c) The number of available host nests is fixed and a host bird can discover an alien egg with a probability of $P_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest to build a completely new nest in a new location.

Cuckoo search algorithm initially requires the selection of a population of n eggs each of which represents a potential solution to the design problem under consideration. This means that it is necessary to generate n solution vector of $\{x\} = \{x_1, x_2, \ldots, x_m\}^T$ in a design problem with m decision variables. For each potential solution vector, the value of objective function $f(\boldsymbol{x})$ is also calculated. The algorithm then generates a new solution $\boldsymbol{x}_i^{v+1} = \boldsymbol{x}_i^v + \beta\boldsymbol{\lambda}$ for cuckoo i where $\boldsymbol{x}_i^{v+1}$ and $\boldsymbol{x}_i^v$ are the previous and new solution vectors. $\beta > 1$ is the step size which is selected according to the design problem under consideration. $\boldsymbol{\lambda}$ is the length of step size which is determined according to random walk with Levy flights. A random walk is a stochastic process in which particles or waves travel along random trajectories which consist of taking successive random steps. The search path of a foraging animal can be modeled as random walk. A Levy flight is a random walk in which the steps are defined in terms of the step lengths which have a certain probability

distribution, with the directions of the steps being isotropic and random. Hence, Levy flights necessitate selection of a random direction and generation of steps under chosen Levy distribution.

The algorithm given in [36] which is called Mantegna algorithm is one of the fast and accurate algorithms which generates a stochastic variable whose probability density is close to Levy stable distribution characterized by arbitrary chosen control parameter α ($0.3 \leq \alpha \leq 1.99$). Using the Mantegna algorithm, the step size **λ** is calculated as

$$\lambda = \frac{x}{|y|^{1/\alpha}} \tag{30}$$

where x and y are the two normal stochastic variables with standard deviation σ_x and σ_y which are given as

$$\sigma_x(\alpha) = \left[\frac{\Gamma(1+\alpha)\sin(\pi\alpha/2)}{\Gamma((1+\alpha)/2)\alpha 2^{(\alpha-1)/2}}\right]^{1/\alpha} \text{ and } \sigma_y(\alpha) = 1 \text{ for } \alpha = 1.5 \tag{31}$$

in which the capital Greek letter Γ represents the Gamma function ($\Gamma(z) = \int_0^\infty t^{z-1}e^{-z}\mathrm{d}t$) that is the extension of the factorial function with its argument shifted down by 1 to real and complex numbers. If $z = k$ is a positive integer $\Gamma(k) = (k-1)!$.

The steps of the cuckoo search algorithm are as follows:

1. Select values for cuckoo search algorithm parameters which are the number of nests (eggs) (n), the step size parameter (β), discovering probability (p_a), and maximum number of iterations for termination of the cycles.
2. Generate initial population of n host nests $\{x_i\}, (i = 1, 2, \ldots, n)$ randomly each of which represents a candidate solution to the optimization problem with objective function of $f(x)$ and decision variables $\{x\} = \{x_1, x_2, \ldots, x_m\}^T$.
3. Get a cuckoo randomly by Levy flights using $\boldsymbol{x}_i^{y+1} = \boldsymbol{x}_i^y + \beta\boldsymbol{\lambda}$ and evaluate its fitness F_i. Here λ is a random walk based on Levy flights which is calculated from (30) to (31).
4. Choose randomly a nest among n (say j) and evaluate its fitness F_j. If $Fj < F_i$, replace j by the new solution.
5. Abandon a fraction of worst nests and built new ones. This is carried out depending on p_a probability parameter. First find out whether each nest keeps its current position using Eq. (32). **R** matrix stores 0 and 1 values such that anyone of them is assigned to each component of ith nest, in which 0 means that current position is kept and 1 implies that the current position is to be updated.

$$\mathbf{R}_i \leftarrow \begin{cases} 1 & \text{if} \quad \text{rand} < \text{pa} \\ 0 & \text{if} \quad \text{rand} \geq \text{pa} \end{cases} \tag{32}$$

New nests are conducted by means of Eq. (29).

$$x_i^{t+1} = x_i^t + r \times \mathbf{R}_i \times (\text{Perm1}_i - \text{Perm2}_i) \tag{33}$$

where r is random number between 0 and 1, Perm1 and Perm2 are two row permutations of the corresponding nest, and **R** defines the probability matrix.

6. Rank solutions and find the current best one.
7. Repeat steps 3–6 until the termination criterion is satisfied which is usually taken as the maximum number of iterations.

4 Design Examples

The solution algorithms presented above for the metaheuristics are performed for two design examples. These are 568-member unbraced steel space frame and 354-bar steel-braced dome. In all the examples, the design history graphs are plotted, demonstrating the improvement of the feasible best design during search process with all the techniques in their best runs. The number of iterations is taken as 50,000 to make sure that all the techniques are given an equal opportunity to be able to find the global optimum. In all the design examples, the following material properties of the steel are used: modulus of elasticity (E) = 208 GPa (30,167.84 ksi) and yield stress (F_y) = 250 MPa (36.26 ksi). The values of parameters that are required to be initially selected for swarm intelligence-based metaheuristic techniques for both design examples are tabulated in Table 2.

Table 2 Search parameters of optimization algorithms

Algorithm	568-member unbraced space steel frame	354-bar steel-braced dome
ABC	Total number of bees = 50 Maximum cycle number = 1000, Limiting value for number of cycles to abandon food source = 250	Total number of bees = 100 Maximum cycle number = 500 Limiting value for number of cycles to abandon food source = 100
ACO	Number of ants = 100, Maximum number of cycles = 500, β = 0.40, Minimum local update coefficient = 0.7	Number of ants = 50, Maximum number of cycles = 1000, β = 0.20 Minimum local update coefficient = 0.8
CSA	Number of nests = 100, pa = 0.35	Number of nests = 50, pa = 0.90
FFA	Number of fireflies = 25, β = 1.0	Number of fireflies = 30, β = 1.0
PSO	Number of particles = 60 c_1 and c_2 = 1.8	Number of particles = 50 c_1 and c_2 = 1.5

4.1 568-Member Unbraced Space Steel Frame

The first design example is selected as a three-dimensional, ten-story, four-bay steel frame having 220 joints and 568 members which are collected in 25 independent design variables [22]. Figure 2a–c shows 3-D, elevation, and plan views of this frame, and member grouping details are presented in Fig. 2d. The wide-flange (*W*) profile list consisting of 272 ready sections is used to size beam and column members [1]. Inner roof beams, outer roof beams, inner floor beams, and outer floor beams of this frame are subjected to 14.72, 7.36, 21.43, and 10.72 kN/m vertical loads, respectively. The un-factored lateral wind loads of this frame are given in Table 3. The drift ratio limits of this frame are defined as 0.914 cm for inter-story drift and 9.14 cm for top story drift. Maximum deflection of beam members is restricted as 1.69 cm.

The final designs and the cross-sectional designations for 25 member groups obtained by each of swarm intelligence-based metaheuristic techniques are given in Table 4. It is obvious from the results that ABC algorithm has obtained the frame with the least weight which is 1,852.1 kN (188,862.34 kg). The second lightest frame with a design weight of 1,899.3 kN (193,675.42 kg) is achieved by ACO algorithm, which is 2.55 % heavier than the lightest frame. The final design attained by PSO algorithm is the third lightest frame which is 1,922.4 kN (196,030.97 kg); 3.8 % heavier than the lightest design. The CSA and FFA algorithms locate optimum solutions that are 1,984.7 kN (202,383.83 kg) and 2,089.8 kN (213,101.09 kg) which are 7.16 and 12.83 %, heavier than the lightest one, respectively. From these results, it is apparent that FFA algorithm has attained the heaviest design. Moreover, from Table 4 it can be concluded that the all constraints are almost at their upper bounds and both drift and strength constraints are dominant in the optimization process for all the algorithms. The convergence history of each algorithm is demonstrated in Fig. 3. The ABC method shows a good convergence rate and comes close to the optimum design nearly after 20,000 structural analyses. PSO algorithm also demonstrates much better convergence rate than the other algorithms. Another interesting result that can be concluded from Fig. 3 is that although FFA exhibits relatively rapid convergence performance to reach to the optimum solution at early stages of design optimization, toward to the final stages its performance deteriorates and it ends up with a local optimum.

4.2 354-Bar Steel-Braced Dome

Steel-braced dome with 40 m (131.23 ft) diameter and a height of 8.28 m (27.17 ft) is taken as a second design example [21]. Braced dome, designed for covering the top of an auditorium at an elevation of 10 m (32.8 ft), consists of 127 joints and 354 members and its plan, elevation, and 3-D views are shown in Fig. 4. The dome is modeled as pin-jointed frame. Dome's 354 members are grouped into 22

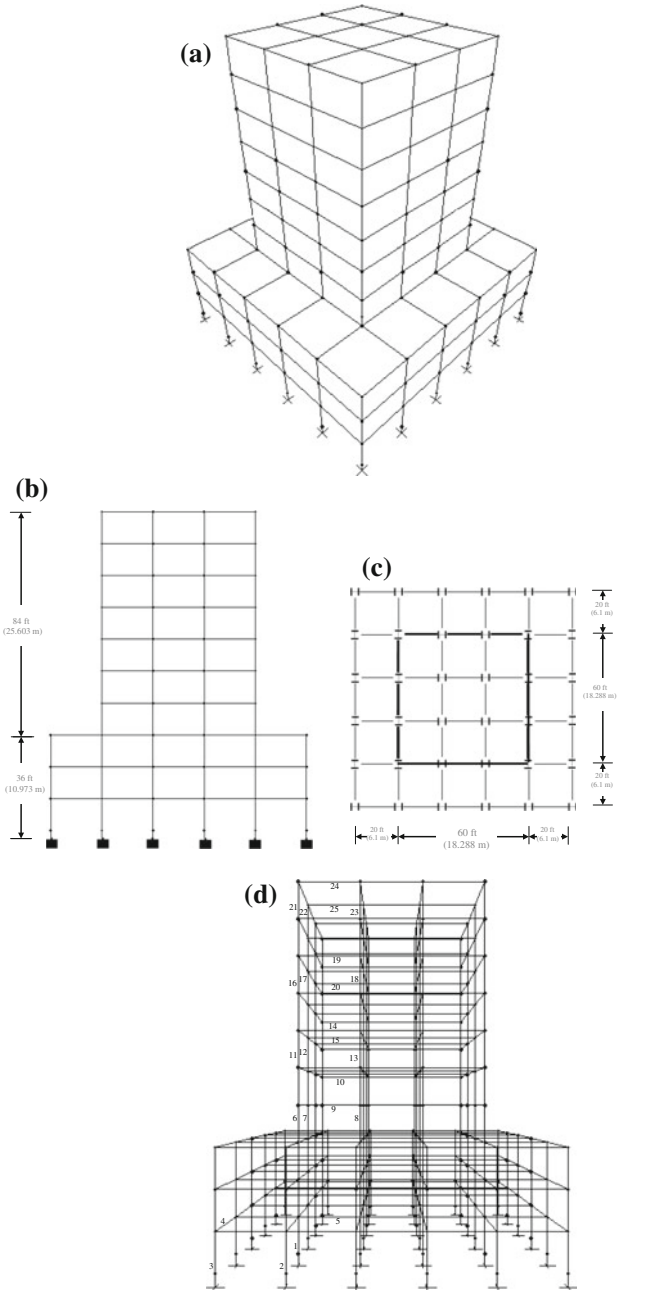

Fig. 2 568-member unbraced space steel frame. **a** 3D view. **b** Elevation view. **c** Plan view. **d** Member grouping

Table 3 Wind loading on 568-member unbraced space steel frame

Floor No	Windward		Leeward	
	(lb/ft)	(kN/m)	(lb/ft)	(kN/m)
1	12.51	0.1825	127.38	1.8585
2	28.68	0.4184	127.38	1.8585
3	44.68	0.6519	127.38	1.8585
4	156.86	2.2886	127.38	1.8585
5	167.19	2.4393	127.38	1.8585
6	176.13	2.5698	127.38	1.8585
7	184.06	2.6854	127.38	1.8585
8	191.21	2.7897	127.38	1.8585
9	197.76	2.8853	127.38	1.8585
10	101.90	1.5743	127.38	1.8585

independent design variables as shown in Fig. 4b, and these design variables are selected from a set of 37 circular hollow steel profile section lists in LRFD-AISC [1]. Three load cases, various combinations of dead (D), snow (S), and wind (W) loads, are applied to the braced dome and these loads are calculated according to ASCE 7-98 specification [37]. These three load cases, (i) D + S, (ii) D + S + W (with negative internal pressure), and (iii) D + S + W (with positive internal pressure), respectively, are shown in Fig. 5. For this example, unbalanced snow loads are not considered and dead and snow loads are assumed like acting on the projected area while wind load acts on the curved surface area. Sandwich-type aluminum material is used for cladding and dead load pressure, including the frame elements used for the girts, is taken as 200 N/m^2. The snow load p_s (kN/m^2) is calculated using the following equation in ASCE 7-98 [37]:

$$p_s = 0.7 C_s C_e C_t I p_g \tag{34}$$

where C_s is the roof slope factor, C_e is the exposure coefficient, C_t is the temperature factor, I is the importance factor, and p_g is the ground snow load. These factors are calculated and/or chosen as follows: $C_s = 1.0$, $C_e = 0.9$, $C_t = 1.0$, $I = 1.1$, and $p_g = 1.1975$ kN/m^2 (25.0 lb/ft^2), resulting in a design snow pressure of $p_s = 830$ N/m^2 (17.325 lb/ft^2).

For the calculation of the design wind pressure p_w, combined effects of internal and external pressures acting on the roof are considered as follows:

$$p_w = q_h G C_p - q_h (G C_{pi}) \tag{35}$$

where q_h is the first the velocity pressure, G is the gust effect factor (taken as 0.85), C_p is the external pressure coefficient, and (GC_{pi}) is the internal pressure coefficient. In this equation, the first term represents the effect of external pressure while the effect of internal pressure is represented by the second term. The braced dome is divided into three regions to compute external wind pressure on it; they are a

Table 4 The optimum designs of 568-member unbraced space steel frame obtained by swarm-based metaheuristic techniques

Group number	ABC	ACO	PSO	CSA	FFA
1	W250X32.7	W250X38.5	W250X44.8	W360X51	W410X38.8
2	W530X85	W200X86	W200X86	W530X85	W460X82
3	W610X174	W610X174	W920X201	W690X170	W840X193
4	W310X23.8	W310X23.8	W250X25.3	W310X23.8	W250X25.3
5	W410X38.8	W410X38.8	W410X38.8	W460X52	W410X46.1
6	W310X107	W690X140	W460X158	W460X106	W530X138
7	W760X196	W920X201	W760X196	W760X196	W310X179
8	W610X174	W840X193	W690X170	W840X176	W920X488
9	W460X106	W460X106	W460X106	W460X106	W530X92
10	W530X150	W610X153	W610X92	W690X152	W840X193
11	W250X101	W460X128	W460X158	W460X106	W460X97
12	W360X122	W460X113	W410X100	W360X134	W310X107
13	W610X125	W690X170	W530X109	W610X174	W690X152
14	W460X60	W360X64	W460X60	W410X60	W530X66
15	W530X85	W250X89	W530X85	W530X85	W410X53
16	W250X80	W360X122	W310X158	W460X68	W460X97
17	W360X122	W410X114	W410X100	W360X134	W310X107
18	W530X85	W610X92	W530X85	W530X85	W610X101
19	W310X32.7	W410X38.8	W310X38.7	W360X32.9	W530X66
20	W410X46.1	W410X46.1	W410X46.1	W410X46.1	W460X68
21	W250X80	W360X72	W310X158	W200X59	W310X74
22	W200X52	W200X71	W250X49.1	W360X134	W310X60
23	W410X60	W460X60	W360X79	W200X59	W460X60
24	W310X38.7	W310X28.3	W200X35.9	W250X32.7	W200X41.7
25	W250X32.7	W310X28.3	W250X38.5	W360X32.9	W460X60
Minimum weight (kN)	1852.1	1899.3	1922.4	1984.7	2089.8
Maximum top story drift (cm)	7.98	7.85	8.02	7.07	6.45
Maximum inter-story drift (cm)	0.913	0.901	0.913	0.859	0.803
Maximum strength constraint ratio	0.925	0.991	0.993	0.903	0.983
Maximum number of iterations	50,000	50,000	50,000	50,000	50,000

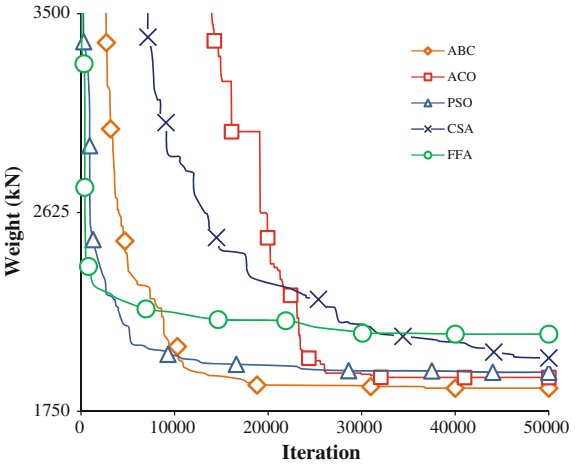

Fig. 3 Search histories of best designs of each swarm-based metaheuristic technique in the design optimization of 568-member unbraced steel space frame

windward quarter, a center half, and a leeward quarter. The external pressure coefficients C_p for each region are calculated separately by considering rise-to-span ratio of the dome as follows: for windward quarter C_p = 0.0105, for center half C_p = –0.907, and for leeward quarter C_p = –0.5. The internal pressure coefficient GC_{pi} is taken as –0.18 and +0.18 in the second and third load cases over the entire internal surface to take into account the suction and uplift effects of the internal pressure, respectively. The first the velocity pressure q_h (N/m^2) evaluated at mean roof height is calculated using the following equation in ASCE 7-98 [37]:

$$q_h = 0.613 K_z K_{zt} K_d V^2 I \tag{36}$$

where K_z is the velocity exposure coefficient (K_z = 1.07 for a mean roof height of 14.14 m (46.4 ft)), K_{zt} is the topographic factor (K_{zt} = 1.087), K_d is the wind directionality factor (K_d = 0.85), V (in m/s) is the basic wind speed (V = 40 m/s (90 mph)), and I is the importance factor (I = 1.15). First the velocity pressure is calculated as 1115 N/m^2 for the braced dome.

The net pressures acting on different regions of the braced domes are calculated by combining internal and external wind pressures as given Eq. (35) and these loadings are shown in Fig. 4. The stress and stability constraints for dome members are imposed according to LRFD-AISC specification [1]. The displacements in any direction are limited to 11.1 cm (4.37 in.) for all nodes.

The minimum weights, maximum constraints values, and pipe section designations of the optimum designs obtained by each swarm-based metaheuristic algorithm are illustrated in Table 5. Search histories of optimization algorithms are shown in Fig. 6. It is clearly seen from the table and the figure that the weight of the best design among all optimum solutions is 142.87 kN (14,568.69 kg) which is obtained by ABC algorithm. This weight is 5.54 % lighter than weight of the best design obtained by CSA, 4.06 % lighter than weight of the best design obtained

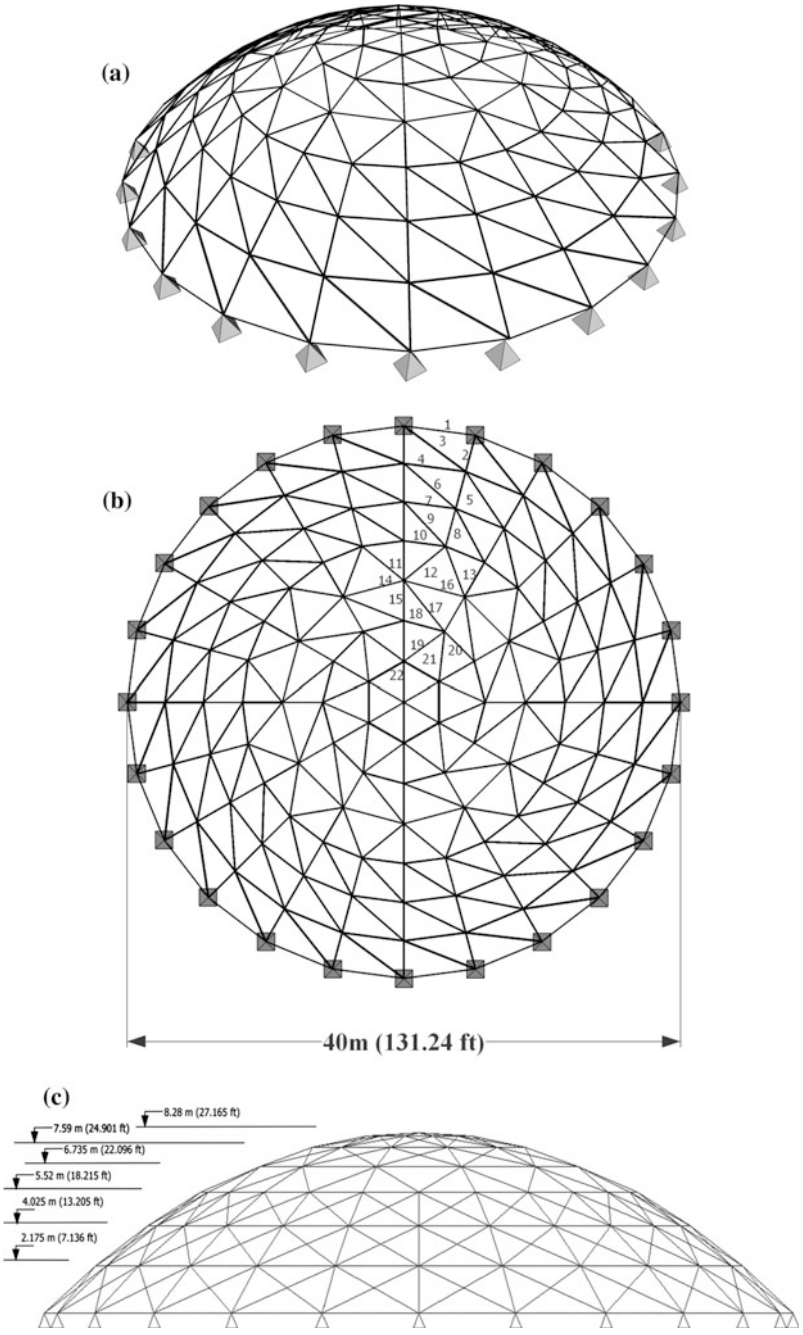

Fig. 4 354-bar steel-braced dome. **a** 3D view. **b** Top view. **c** Side view

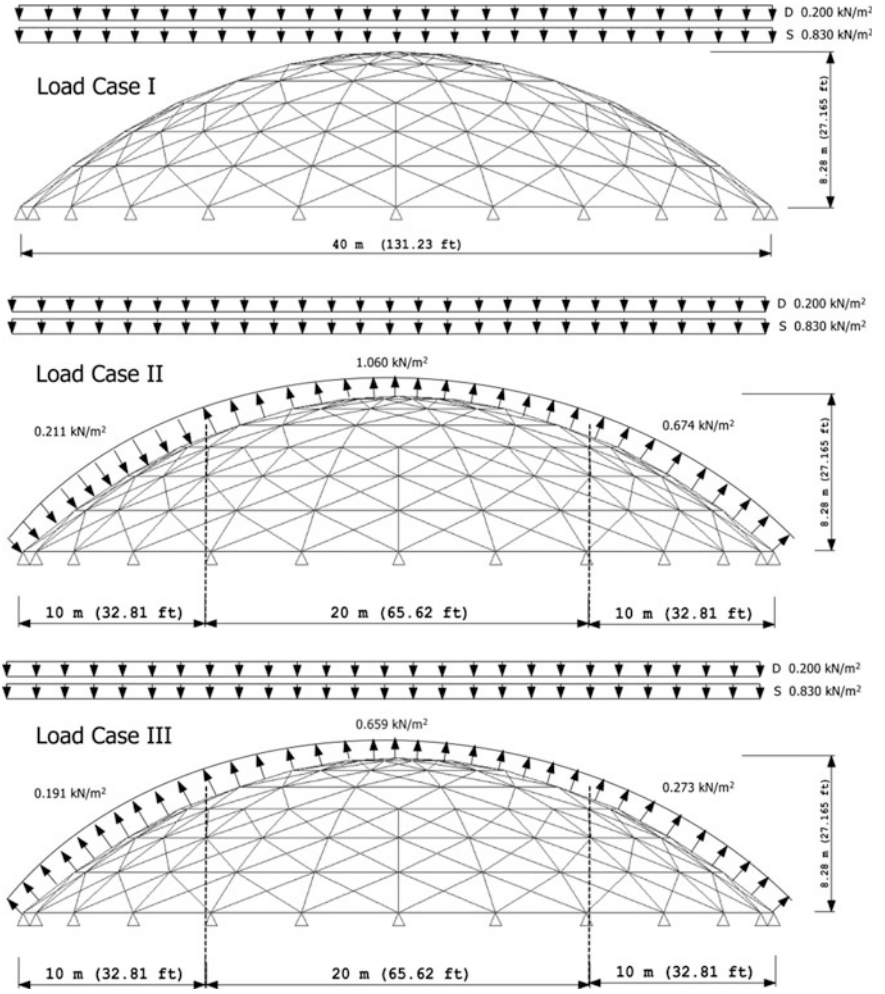

Fig. 5 Loading on the design of 354-bar steel-braced dome

using FFA, 2.65 % lighter than weight of the best design obtained by ACO algorithm, and 1.16 % lighter than weight of best design obtained by PSO algorithm. Weights of the second and the third best designs which are obtained from PSO and ACO are closer to each other. There is only 1.47 % weight difference between these designs. Among all of the five metaheuristic algorithms considered in this study, CSA has shown the worst performance. The minimum weight of the optimum design obtained by this algorithm is 150.78 kN (15,373.34 kg). It is clear that this design is trapped in local optimum.

Table 5 Optimum designs for 354-bar steel-braced dome

Group number	ABC	PSO	ACO	FFA	CSA
1	P2	P2	P2	P2	P2
2	P3	P3	P3	P4	P4
3	P4	P3.5	P4	P3.5	P3.5
4	P3	P3.5	P3.5	P3.5	P3.5
5	P3	P3	P3	P3.5	P3.5
6	P3	P3	P3	P3	P3
7	P3	P3	P3	P3	P3
8	P2.5	P3	P2.5	P2.5	P3
9	P3	P3	P3	P3	P3
10	P3	P3	P3	P3	P3
11	P2.5	P2.5	P2.5	P2.5	P2.5
12	P2.5	P2.5	P2.5	P2.5	P2.5
13	P2.5	P2.5	P2.5	P2.5	P2.5
14	P2.5	P2.5	P2.5	P2.5	P2.5
15	P2.5	P2.5	XP2.5	P2.5	P2.5
16	P2.5	P2.5	P2.5	P2.5	P2.5
17	XP2	XP2	XP2	XP2	XP2
18	P2	P2	XP2	XP2	P2.5
19	XP2	XP2	P2	XP2	XP2
20	P2	P2.5	XP2	XP2	P2.5
21	P2	P2	P2	XP2	P2
22	P2	P2	P2	P2	P2
Minimum weight(kN)	142.87	144.53	146.65	148.67	150.78
Maximum strength ratio	0.997	0.844	0.896	0.906	0.801
Maximum displacement (cm)	1.73	1.67	1.71	1.65	1.61
Maximum no. of iterations	50,000	50,000	50,000	50,000	50,000

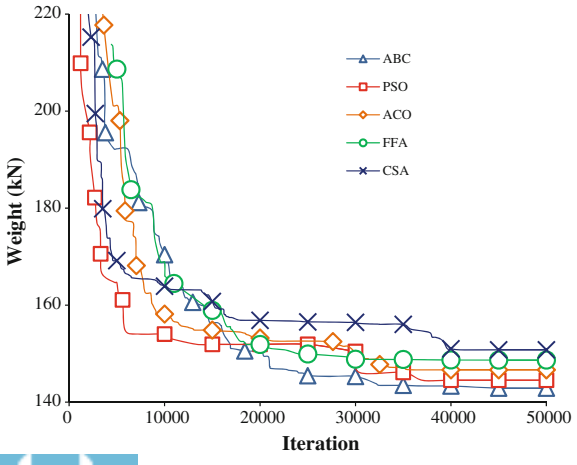

Fig. 6 Search histories of best designs of each swarm-based metaheuristic technique in the design optimization of 354-bar steel-braced dome

5 Summary and Conclusions

Structural engineers are required to design structures optimally due to the diminishing resources and economic competition. The optimum design of a steel frame necessitates selection of steel profiles for its members from the available steel section list such that the behavior of the frame under the external loads is within the limitations described by the steel design codes such as LRFD-AISC and its weight or cost is the minimum. The mathematical modeling of this decision making problem turns out to be discrete nonlinear programming problem. Recent studies are shown that metaheuristic techniques are more suitable to attain the optimum design of real size steel frames under the steel design code provisions than mathematical programming algorithms. In this study, five optimum structural design algorithms are developed each of which is based on one of the recent swarm intelligence-based metaheuristic techniques. These techniques are selected as ant colony optimization, particle swarm optimizer, artificial bee colony algorithm, firefly algorithm, and cuckoo search algorithm, respectively. It is shown that all these are robust and can effectively obtain the optimum designs under the real loading conditions as well as code provisions. However, their performance differs in approaching to the global optimum. The experience obtained from two real size steel frames, one is 568-member steel space frame and the other 354-bar steel-braced dome, has shown that among the five swarm intelligence techniques, artificial bee colony algorithm has reached the e lightest optimum designs. The minimum weights determined by the artificial bee colony algorithm were 12.83 % in the case of steel space frame and 5.54 % in the case of steel-braced dome lighter than the heaviest optimum design, respectively. The heaviest optimum designs are obtained by cuckoo search algorithm and firefly algorithm, respectively. One reason for this good performance may be having only two parameters to be selected by users which are the total number of bees and the limiting number of cycles to abandon food sources. Particle swarm optimizer and cuckoo search algorithms are also similar to artificial bee colony algorithm regarding the number of parameters. These techniques have only two parameters to be initialized. This may be the reason why the optimum designs attained by the particle swarm optimizer and cuckoo search algorithm are close to the one determined by the artificial bee colony algorithm. Having several parameters is a disadvantage due to the fact that these parameters are required to be tuned properly for an efficient performance. Finding the best combination of these parameters becomes another optimization problem to be solved. Inspection of the search histories indicates the fact that particle swarm optimizer demonstrates the second best behavior while firefly algorithm is trapped in a local optimum. Cuckoo search and ant colony algorithms exhibit slow convergence in the initial stages of the search but toward the final stages they recover and attain optimum designs not very far from the one determined by artificial bee colony algorithm. It should be pointed out that the swarm intelligence-based algorithms used in obtaining the solution of two steel frame design examples are coded according to their standard forms given by their originators. No

enhancements are employed that might improve their performance some of which already available in the literature. Although two real size steel frame design examples may not be considered sufficient without carrying out statistical analysis to draw concrete conclusions regarding the efficiency of these five algorithms, one conclusion does certainly be made. Swarm intelligence-based metaheuristic algorithms are robust, easy to program, and provide solutions to complex discrete nonlinear programming problem of structural steel design problems in a better way than mathematical programming techniques.

References

1. Load and Resistance Factor Design (LRFD), Volume 1, Structural Members Specifications Codes, 3rd edn. American Institute of Steel Construction (2001)
2. Rao, S.S.: Engineering Optimization; Theory and Practice, 4th edn. Wiley, New York (2009)
3. Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms, Luniver Press, Bristol (2008)
4. Kochenberger, G.A., Glover, F.: Handbook of Meta-Heuristics. Kluwer Academic Publishers, Dordrecht (2003)
5. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput. Surv. **35**(30), 268–308 (2003)
6. De Castro, L.N., Von Zuben, F.J.: Recent Developments in Biologically Inspired Computing. Idea Group Publishing, Hershey (2005)
7. Dreo, J., Petrowski, A., Siarry, P., Taillard, E.: Meta-Heuristics for Hard Optimization. Springer, Berlin (2006)
8. Gonzales, T.F.: Handbook of Approximation Algorithms and Metaheuristics. Chapman&Hall, CRC Press, London (2007)
9. Luke, S.: Essentials of Metaheuristics. http://cs.gmu.edu/∼sean/book/metaheuristics/ (2010). Accessed May 2015
10. Saka, M.P., Dogan, E., Aydogdu, I.: Review and analysis of swarm-intelligence based algorithms, Chapter 2. In: Yang, Cui, Xiao, Gandomi (eds.) Swarm Intelligence and Bio-Inspired Computation, Theory and Applications. Elsevier, Amsterdam. ISBN: 978-0-12-405163-8 (2013)
11. Saka, M.P.: Optimum design of skeletal structures: a review, Chapter 10. In: Topping, B.H.V. (ed.) Progress in Civil and Structural Engineering Computing, pp. 237–284. Saxe-Coburg Publications, Edinburgh (2003)
12. Saka, M.P.: Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review, Chapter 6. In: Topping, B.H.V. (ed.) Civil Engineering Computations: Tools and Techniques, pp. 105–147. Saxe-Coburgh Publications, Edinburgh (2007)
13. Lamberti, L., Pappalettere, C.: Metaheuristic design optimization of skeletal structures: a review. Comput. Technol. Rev. **4**, 1–32 (2011)
14. Saka, M.P.: Recent developments in metaheuristic algorithms: a review. Comput. Technol. Rev. **5**, 31–78 (2012)
15. Saka, M.P., Geem, Z.W.: Mathematical and metaheuristic applications in design optimization of steel frame structures: an extensive review. Math. Probl. Eng. (2013)
16. Saka, M.P.: Shape and topology optimization design of skeletal structures using metaheuristic algorithms: a review. Comput. Technol. Rev. **9**, 31–68 (2014)
17. Ad Hoc Committee on Serviceability: Structural serviceability: a critical appraisal and research needs. J. Struct. Eng. ASCE **112**(12), 2646–2664 (1986)
18. Ellingwood, B.: Serviceability guidelines for steel structures. Eng. J. AISC **26**, 1–8 (1989)

19. Chen, W.F., Kim, S.-E.: LRFD Steel Design Using Advanced Analysis. CRC Press, Boca Raton (1997)
20. ASCE 7-05: Minimum Design Loads for Building and Other Structures (2005)
21. Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., Saka, M.P.: Performance evaluation of metaheuristic techniques in the optimum design of real size pin jointed structures. Comput. Struct. **87**(5–6), 284–302 (2009)
22. Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., Saka, M.P.: Comparison of non-deterministic search techniques in the optimum design of real size steel frames. Comput. Struct. **88**(17–18), 1033–1048 (2010)
23. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by Ant Colony. In: Proceedings of First European Conference on Artificial Life, U.S.A., pp. 134–142 (1991)
24. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. thesis, Dipartimento Elettronica e Informazione, Politecnico di Milano, Italy (1992)
25. Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Book, Massachusetts Institute of Technology, U.S.A. (2004)
26. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE Press (1995)
27. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
28. Karaboga, D.: An idea based on Honey Bee Swarm for numerical optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
29. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) Algorithm. J. Global Optim. **39**(3), 459–471 (2007)
30. Yang, X.-S.: Firefly algorithms for multimodal optimization, chapter in Stochastic algorithms: foundations and applications. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009, Lecture Notes in Computer Science, vol. 5792, pp. 169–178 (2009)
31. Yang, X.-S., He, X.: Firefly algorithm: recent advances and applications. Int. J. Swarm Intell. **1**(1), 36–50 (2013)
32. Gandomi, A.H., Yang, X.-S., Alavi, A.H.: Mixed variable structural optimization using Firefly algorithm. Comput. Struct. **89**(23–24), 2325–2336 (2011)
33. Yang, X.-S., Deb, S.: Engineering optimization by Cuckoo Search. Int. J. Math. Model. Numer. Optim. **1**(4), 330–343 (2010)
34. Coello, C.A.C.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput. Methods Appl. Mech. Eng. **191**, 1245–1287 (2002)
35. Carbas, S., Saka, M.P.: Efficiency of improved harmony search algorithm for solving engineering optimization problems. Int. J. Optim. Civil Eng. **3**(1), 99–114 (2013)
36. Mantegna, R.N.: Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. Phys. Rev. **49**(5), 4677–4683 (1994)
37. ASCE 7-98: Minimum Design Loads for Buildings and Other Structures, American Society of Civil Engineers (1998)

Metaheuristic Optimization in Structural Engineering

S.O. Degertekin and Zong Woo Geem

Abstract Metaheuristic search methods have been extensively used for optimization of the structures over the past two decades. Genetic algorithms (GA), ant colony optimization (ACO), particle swarm optimization (PSO), harmony search (HS), big bang-big crunch (BB-BC), artificial bee colony algorithm (ABC) and teaching–learning-based optimization (TLBO) are the most popular metaheuristic optimization methods. The basic principle of these methods is that they make an analogy between the natural phenomena and the optimization problems. In this chapter, recently developed metaheuristic optimization methods such as self-adaptive harmony search and teaching–learning-based optimization are reviewed and the performance of these methods in the field of structural engineering are compared with each other and the other metaheuristic methods.

Keywords Metaheuristic optimization · Structural engineering · Truss structures

1 Introduction

A family of metaheuristic optimization methods based on swarm intelligence have been developed in the past two decades. These methods simulate the behaviour of different groups/swarms/colonies of animals and insects. Ant colony optimization (ACO), harmony search (HS), particle swarm optimization (PSO), big bang-big crunch optimization (BB-BC) and artificial bee colony optimization (ABC), teaching–learning-based optimization (TLBO) are few examples of recent metaheuristic

S.O. Degertekin
Department of Civil Engineering, Dicle University,
21280 Diyarbakir, Turkey

Z.W. Geem (✉)
Department of Energy IT, Gachon University,
Seongnam 461-701, South Korea
e-mail: zwgeem@gmail.com

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*,
Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_4

algorithms. They also are classified as population-based or nature-inspired optimization methods. The main philosophy of all metaheuristic optimization methods is the mimicking of the natural phenomenon. Design optimization of skeletal structures using metaheuristic search methods is an important field of engineering under continuous development. The state-of-the-art utilization of metaheuristic algorithms in weight or cost optimization of skeletal structures have been recently reviewed by Lamberti and Pappalettere [1] and Saka [2].

The ACO was originally proposed by Dorigo et al. [3] for optimization problems. The method simulates the foraging behaviour of real-life ant colonies. The ACO attempts to model some of the fundamental capabilities observed in the behaviour of ants as a method stochastic combinatorial optimization [4]. In addition to its different applications, the method has also been used for design optimization of structural systems. ACO was used for optimization of truss structures by Camp and Bichon [5], Capriles et al. [6], Serra and Venini [7], and Hasancebi et al. [8], and frame structures by Camp et al. [4], Kaveh and Shojaee [9], Hasancebi et al. [10], Kaveh and Talatahari [11].

HS was first developed by Geem et al. [12] for solving combinatorial optimization problems. The method bases on the analogy between the musical process of searching for a perfect state of harmony and searching for solutions to optimization problems. HS has been used for a variety of structural optimization problems including optimum design of truss structures [8, 13, 14], geodesic domes [15], grillage systems [16] and steel frames [17–19].

In recent years, improved/modified HS algorithms have been developed in order to increase the efficiency of the method. Saka and Hasancebi [20] developed an adaptive harmony search algorithm for design code optimization of steel structures. Hasancebi et al. [21] proposed an adaptive harmony search method for structural optimization. Lamberti and Pappalettere [22] proposed an improved harmony-search algorithm, where trial designs are generated including information on the gradients of cost function for truss structure optimization. Two improved harmony search algorithms called efficient harmony-search (EHS) algorithm and self-adaptive harmony-search (SAHS) algorithm were proposed by Degertekin [23] for sizing the optimization of truss structures.

The PSO method was first developed by Kennedy and Eberhart [24]. It is based on the premise that social sharing of information among members of a species offers and evolutionary advantage [25]. PSO has been used in optimization of skeletal structures [26–29]. Researchers introduced new features in the standard implementation of PSO. Li et al. [28, 29] developed a heuristic particle swarm optimizer (HPSO), which combines the PSO scheme and the HS scheme, for sizing optimization of truss structures. Kaveh and Talatahari [30, 31] introduced a particle swarmer, ant colony optimization and harmony search scheme for truss structures with both discrete [30] and continuous variables [31].

The BB-BC proposed by Erol and Eksin [32] simulates the theories of the evolution of the universe. According to this theory, in the big bang phase energy dissipation produces disorder, and randomness is the main feature of this phase; whereas, in the big crunch phase, randomly distributed particles are drawn into an

order [33]. BB-BC algorithm was applied for sizing optimization of truss structures [34]. In order to improve the convergence capability of standard BB-BC algorithm, Kaveh and Talatahari [33, 35] developed hybrid BB-BC (HBB-BC) algorithm to optimize space trusses and ribbed domes.

The ABC method was first developed by Karaboga [36] for numerical function optimization. The ABC is an optimization method based on the intelligent behaviour of honey bee swarm. The ABC has successfully been applied to the size optimization of truss structures with both continuous [37] and discrete variables [38].

Another metaheuristic method called 'teaching-learning-based optimization (TLBO)' has been proposed by Rao et al. [39] for constrained mechanical design optimization problems. The method bases on the effect of influence of a teacher on learners and the effect of learners with each other. Rao et al. [40] developed the TLBO method for large-scale nonlinear optimization problems for finding global solutions. TLBO was employed for optimum design of planar steel frames [41] and sizing optimization of truss structures Degertekin and Hayalioglu [42].

In this chapter, the robustness of the SAHS [23] and TLBO [42] will be investigated in the optimization of truss type structures. Three benchmark truss structures existing in the current literature are presented to test the efficiency of the methods. The results obtained from these methods will be compared with those of other metaheuristic optimization algorithms recently presented in the literatures like particle swarm optimization (PSO), heuristic particle swarm optimizer (HPSO), hybrid particle swarm optimization (HPSO), big bang-big crunch optimization (BB-BC), heuristic particle swarm ant colony optimization (HPSACO), hybrid big bang-big crunch optimization (HBB-BC), artificial bee colony optimization (ABC-AP) and improved harmony search algorithm (IHS).

The rest of this study is organized as follows. The formulation of optimum design problem is given in Sect. 2. SAHS and TLBO methods are explained in Sects. 3 and 4. The results obtained from the design examples are presented and compared with other metaheuristic optimization methods in Sect. 5. Finally, conclusions are presented in Sect. 6.

2 Formulation of Optimum Design Problem

The minimum weight design problem for a truss structure can be formulated as

Find $X = [x_1, x_2, \ldots, x_{ng}]$ to minimize

$$W(X) = \sum_{k=1}^{ng} x_k \sum_{i=1}^{mk} \rho_i L_i \tag{1}$$

subject to the following normalized constraints

$$g_{\mathrm{nl}}^{s}(X) = \frac{|\sigma_{\mathrm{nl}}|}{|\sigma_{\mathrm{nu}}|} - 1 \leq 0, \quad 1 \leq n \leq \mathrm{nm}, \quad 1 \leq l \leq \mathrm{nl} \tag{2}$$

$$g_{\mathrm{nl}}^{b}(X) = \frac{|\sigma_{\mathrm{cl}}|}{|\sigma_{\mathrm{cu}}|} - 1 \leq 0, \quad 1 \leq n \leq \mathrm{ncm}, \quad 1 \leq l \leq \mathrm{nl} \tag{3}$$

$$g_{jl}^{d}(X) = \frac{|d_{jl}|}{|d_{ju}|} - 1 \leq 0, \quad 1 \leq j \leq \mathrm{nn}, \quad 1 \leq l \leq \mathrm{nl} \tag{4}$$

$$x_{\min} \leq x_k \leq x_{\max}, \quad k = 1, 2, \ldots \mathrm{ng} \tag{5}$$

where X is the vector containing the design variables, $W(X)$ is the weight of the truss structure, ng is the total number of member groups (i.e. design variables), x_k is the cross-sectional area of the members belonging to the group k, mk is the total number of members in the group k, ρ_i is the density of member i, L_i is the length of member i, $g_{\mathrm{nl}}^{s}(X)$, $g_{\mathrm{nl}}^{b}(X)$ and $g_{jl}^{d}(X)$ are the constraint violations for member stress, member buckling stress and joint displacements of the structure. σ_{nl} and σ_{cl} are the member stress and the member buckling stress of the nth member due to loading condition l, σ_{nu} and σ_{cu} are their upper limits. d_{jl} is the nodal displacement of the jth translational degree of freedom due to loading condition l, d_{ju} is its upper limit. nl is the number of load conditions, nn is the number of nodes, max and min are the upper and lower limits for cross-sectional area.

The optimum design of truss structures must satisfy optimization constraints stated by Eqs. (2)–(5). In this study, the constraints are handled using a modified feasible-based mechanism [30]. The efficiency of the method was previously verified for optimization of truss structures [23, 30]. The method consists of the following four rules [30]:

Rule 1: Any feasible design is preferred to any infeasible design.

Rule 2: Infeasible designs containing slight violation of the constraints (from 0.01 in the first search to 0.001 in the last search) are considered as feasible designs.

Rule 3: Between two feasible designs, the one having the better objective function value is preferred.

Rule 4: Between two infeasible designs, the one having the smaller constraint violation is preferred.

3 Self-Adaptive Harmony Search Algorithm (SAHS)

Harmony-search (HS) algorithm for sizing optimization of the truss structures could be explained in the following steps [23]:

Step 1. Initializing the harmony search parameters
HS parameters are assigned in this step. The number of design vectors in harmony memory (HMS), harmony memory consideration rate (HMCR), pitch adjusting rate (PAR) and the stopping criterion are selected in this step.

Step 2. Initializing harmony memory
All design vectors are stored in the harmony memory (HM). The HM matrix given in Eq. (6) is filled with randomly generated design vectors as the size of the harmony memory (HMS) in this step.

$$\mathrm{HM}=\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{ng-1}^1 & x_{ng}^1 \\ x_1^2 & x_2^2 & \cdots & x_{ng-1}^2 & x_{ng}^2 \\ \vdots & \vdots & ::: & \vdots & \vdots \\ \vdots & \vdots & ::: & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{ng-1}^{HMS-1} & x_{ng}^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{ng-1}^{HMS} & x_{ng}^{HMS} \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \begin{matrix} W(X^1) \\ W(X^2) \\ \vdots \\ \vdots \\ W(X^{HMS-1}) \\ W(X^{HMS}) \end{matrix} \tag{6}$$

In the HM, each row represents a truss design. $X^1, X^2,\ldots, X^{HMS-1}, X^{HMS}$ and $W(X^1)$, $W(X^2),\ldots,\ W(X^{HMS-1}),\ W(X^{HMS})$ are designs and the corresponding objective function values, respectively. The truss designs in the HM are sorted by their objective function values $(W(X^1) \le W(X^2) \le \cdots \le W(X^{HMS-1}) \le W(X^{HMS}))$ which are calculated using Eq. (1).

Step 3. Improvising a new harmony
A new harmony (i.e. new truss design) $X^{new} = (x_1^{new}, x_2^{new}, \ldots, x_{ng}^{new})$ is generated using three rules: (i) HM consideration, (ii) pitch adjustment and (iii) random generation. Generating a new harmony is called 'improvisation' [13].

In the HM consideration, the value of the first design variable x_1^{new} for the new harmony is chosen from the HM, (i.e. $\{x_1^1, x_1^2, \ldots.., x_1^{HMS-1}, x_1^{HMS}\}$) or from the possible range of values. The other design variables of new harmony $(x_2^{new}, \ldots, x_{ng-1}^{new}, x_{ng}^{new})$ are chosen by the same consideration. HMCR is applied as follows:

$$\begin{cases} x_i^{new} \in \{x_i^1, x_i^2, \ldots\ldots, x_i^{HMS-1}, x_i^{HMS}\} & \text{with probability HMCR} \\ x_i^{new} \in X_s & \text{with probability } (1-\mathrm{HMCR}) \end{cases} \tag{7}$$

where X_s is the set of the possible range of values for each design variable $(x_{min} \le X_s \le x_{max})$. The HMCR, which varies between 0 and 1, is the rate of

choosing one value from historical values stored in the HM, while (1–HMCR) is the rate of randomly selecting one value from the possible range of values [14]. For example, a HMCR value of 0.95 indicates that HS algorithm will choose the design variable from historically stored values in the HM with a 95 % probability and from the entire possible range with a 5 % probability [13].

Any design variable of the new harmony obtained by the memory consideration is examined to determine whether it is pitch-adjusted or not. This is performed by the pitch adjusting rate (PAR). PAR investigates a better design in the neighbouring of the current design and is applied as follows:

Pitch adjusting decision for

$$x_i^{\text{new}} \leftarrow \begin{cases} \text{yes with probability} & \text{PAR} \\ \text{no with probability} & (1 - \text{PAR}) \end{cases} \tag{8}$$

The value of (1–PAR) sets the rate of doing nothing, whereas the value of PAR indicates that x_i^{new} is replaced as follows:

$$x_i^{\text{new}} \leftarrow x_i^{\text{new}} + \text{bw} \times u(-1, 1) \tag{9}$$

where bw is the arbitrary distance bandwidth for continuous variable and $u(-1, 1)$ is a uniform distribution between -1 and 1. For example, a PAR of 0.1 indicates that the algorithm will choose a neighbouring value with 10 % × HMCR probability [13]. HMCR and PAR parameters are introduced to allow the solution to escape from the local optima and to improve the global optimum prediction of HS algorithm [13].

Step 4. Updating the harmony memory

If the new harmony $X^{\text{new}} = (x_1^{\text{new}}, x_2^{\text{new}}, \ldots, x_{\text{ng}}^{\text{new}})$ is better than the worst design in the HM, judged in terms of the objective function value, the new harmony is included in the HM and the worst harmony is excluded from the HM. In this process, the HM is sorted again by objective function values.

Step 5. Terminating the process

Steps 3 and 4 are repeated until the termination criterion is satisfied.

The proposed SAHS algorithm differs from the standard HS algorithm as indicated in the following aspects:

SAHS algorithm presented in this study dynamically updates PAR during the search process as follows [23]:

$$\text{PAR}(ns) = \text{PAR}_{\max} - \frac{(\text{PAR}_{\max} - \text{PAR}_{\min})}{NI} \times ns \tag{10}$$

In SAHS algorithm, bw is completely removed and Eq. (9) is replaced as follows:

$$x_i^{\text{new}} = x_i^{\text{new}} + [\max(\text{HM})_i - x_i^{\text{new}}] \times u(0,1) \quad \text{if } u(0,1) \le 0.5 \tag{11}$$

$$x_i^{\text{new}} = x_i^{\text{new}} - [x_i^{\text{new}} - \min(\text{HM})_i] \times u(0,1) \quad \text{if } u(0,1) > 0.5 \tag{12}$$

where $\min(\text{HM})_i$ and $\max(\text{HM})_i$ are the lowest and highest values of the ith design variable in the HM. $u(0,1)$ is a uniform random number in the [0,1] range. Since $\min(\text{HM})_i$ and $\max(\text{HM})_i$ approach optimum gradually, SAHS algorithm produces finer adjustments to the harmony.

4 Teaching–Learning-Based Optimization

The TLBO method presents a mathematical model for optimization problems based on the simple teaching process. In the TLBO, the learners in a class are considered as the population. The teacher is accepted as the well-versed person in his/her profession. Hence, the learner with the highest mark in a class is mimicked as a teacher.

An analogy between the TLBO and the optimization of truss structures is established in the following way: a class is considered as a population which contains truss designs, a learner in a class denotes a truss design in the population, a design variable represents a subject taught to student, the grade of a student denotes the weight of the truss design, the teacher is the truss design with the lowest weight in the population.

Optimization of truss structures using the TLBO method consists of following steps [42]:

Step 1. Initializing the TLBO

In this step, the class is filled with randomly generated learners (truss designs) as the size of the population (ps).

$$\text{ps} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{\text{ng}-1}^1 & x_{\text{ng}}^1 \\ x_1^2 & x_2^2 & \cdots & x_{\text{ng}-1}^2 & x_{\text{ng}}^2 \\ \vdots & \vdots & ::: & \vdots & \vdots \\ \vdots & \vdots & ::: & \vdots & \vdots \\ x_1^{\text{ps}-1} & x_2^{\text{ps}-1} & \cdots & x_{\text{ng}-1}^{\text{ps}-1} & x_{\text{ng}}^{\text{ps}-1} \\ x_1^{ps} & x_2^{\text{ps}} & \cdots & x_{\text{ng}-1}^{\text{ps}} & x_{\text{ng}}^{\text{ps}} \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \begin{matrix} W(X^1) \\ W(X^2) \\ \vdots \\ \vdots \\ W(X^{\text{ps}-1}) \\ W(X^{\text{ps}}) \end{matrix} \tag{13}$$

In the class, each row represents a truss design. X^1, X^2,…, $X^{\text{ps}-1}$, X^{ps} and $W(X^1)$, $W(X^2)$,…, $W(X^{\text{ps}-1})$, $W(X^{\text{ps}})$ are truss designs and the corresponding weight values, respectively. It should be noted that the designs in the class are sorted by their weight values ($W(X^1) \le W(X^2) \le \cdots \le W(X^{\text{ps}-1}) \le W(X^{\text{ps}})$) which are calculated using Eq. (1).

Step 2. Teaching phase

The truss design with the lowest objective function ($W(X^1)$) is assigned as the teacher ($X^{\text{teacher}} = X^1$). The aim of the teacher is to put effort to move the mean of the class (X^{mean}). Therefore, ith design ($i \neq 1$) is modified using the following expression:

$$X^{\text{new},i} = X^i + r(X^1 - T_F X^{\text{mean}}) \tag{14}$$

in which X^i and $X^{\text{new},i}$ are the current and new designs, respectively. r is the random number uniformly distributed in the range of [0,1], T_F is the teaching factor which is either 0 or 1 [39] and X^{mean} is the mean of the designs calculated as the following way:

$$X^{\text{mean}} = \left[m\left(\text{class}\left(\sum_{i=1}^{\text{ps}} x_1^i\right)\right), m\left(\text{class}\left(\sum_{i=1}^{\text{ps}} x_2^i\right)\right), \ldots\ldots, m\left(\text{class}\left(\sum_{i=1}^{\text{ps}} x_{\text{ng}}^i\right)\right)\right] \tag{15}$$

where $m(\cdot)$ is the mean of the design variable. If the new design ($X^{\text{new},i}$) is better than the current design (X^i) (i.e. $W(X^{\text{new},i}) < W(X^i)$), the new design is replaced with the current design, $X^i = X^{\text{new},i}$.

Step 3. Learning phase

In addition to the teacher's effort to improve the mean of the class, the learners also interact with each other to improve themselves. A design in the population is randomly interacted with other designs to improve its quality. The learning phase is applied to learn new information between the design i and j ($i \neq j$) in the population and can be expressed as [41]

$$X^{\text{new},i} = X^i + r(X^i - X^j) \quad \text{if } W(X^i) < W(X^j) \tag{16a}$$

$$X^{\text{new},i} = X^i + r(X^j - X^i) \quad \text{if } W(X^j) < W(X^i) \tag{16b}$$

in which X^j is the randomly determined design which has to be different from X^i. If the value of $W(X^{\text{new},i})$ is better than $W(X^i)$ (i.e. $W(X^{\text{new},i}) < W(X^i)$), the new design is replaced with the current design $X^i = X^{\text{new},i}$.

Step 4. Terminating the search process

The steps 2 and 3 are repeated until the lightest truss design does not improve during a predetermined number of structural analyses.

5 Design Examples

The effectiveness and robustness of the SAHS [23] and TLBO [42] are tested using four truss structures. The results obtained by the methods are compared with those of HS [13], IHS [22], PSO [25], PSO, PSOPC and HPSO [28], HPSACO [31], HBB-BC [33], BB-BC [34] and ABC-AP [37].

SAHS algorithm produces the minimum weight design for the values of 20 for HMS, 0.90 for HMCR, 0.20 and 0.80 for PARmin and PARmax, 0.001 and 0.01 [23].

Two tuning parameters are employed in the TLBO: the population size (ps) and the number of designs generated in the learning phase (ndlp). The best combination of them obtained after sensitivity analysis and the minimum weight design for the TLBO is obtained by setting ps = 30, ndlp = 4 [42].

Twenty independent runs are made for each design example involving 20 different initial designs because of the stochastic nature of the SAHS and TLBO. The best designs obtained by the methods, the number of structural analyses required to the optimum solutions, the average weight and the standard deviation of 20 independent runs are given in the tables. The SAHS and TLBO are coded in FORTRAN language and executed on a Intel Pentium Core 2 Duo 2.2 GHz machine.

5.1 Ten-Bar Plane Truss

The planar ten-bar plane truss, shown in Fig. 1 is the first design example. The Young's modulus and density of truss members are 10^4 ksi (1 ksi = 6.895 MPa) and 0.1 lb/in^3, respectively. The allowable stress for all members is specified as 25 ksi both in tension and compression. The maximum displacements of all free nodes in the X and Y directions are limited to ±2. Each member is considered as a design variable with the minimum gauge of 0.1 in^2.

The results obtained by the SAHS [23], TLBO [42] and the other optimization methods are reported in Table 1. The presented methods in this chapter found the

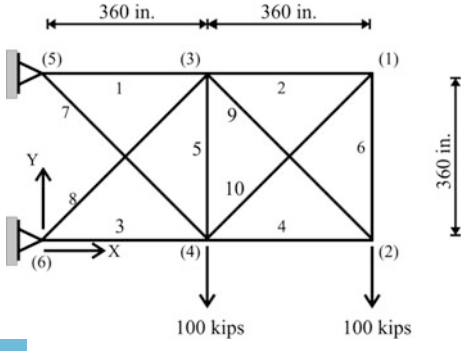

Fig. 1 Ten-bar plane truss (1 in. = 2.54 cm, 1 kip = 4.448 kN)

Table 1 The pseudocode of the HS algorithm

Design variables A_i (in.^{2})	[13]	[25]	[28]			[31]	[22]	[37]	Proposed [23, 42]	
	HS	PSO	PSO	PSOPC	HPSO	HPSACO	IHS	ABC-AP	SAHS	TBLO
A_1	30.15	33.500	33.469	30.569	30.704	30.307	30.5222	30.548	30.394	30.4286
A_2	0.102	0.100	0.110	0.100	0.100	0.100	0.1000	0.100	0.100	0.1000
A_3	22.71	22.766	23.177	22.974	23.167	23.434	23.2005	23.180	23.098	23.2436
A_4	15.27	14.417	15.475	15.148	15.183	15.505	15.2232	15.218	15.491	15.3677
A_5	0.102	0.100	3.649	0.100	0.100	0.100	0.1000	0.100	0.100	0.1000
A_6	0.544	0.100	0.116	0.547	0.551	0.5241	0.5513	0.551	0.529	0.5751
A_7	7.541	7.534	8.328	7.493	7.460	7.4365	7.4572	7.463	7.488	7.4404
A_8	21.56	20.467	23.340	21.159	20.978	21.079	21.0367	21.058	21.189	20.9665
A_9	21.45	20.392	23.014	21.156	21.508	21.229	21.5288	21.501	21.342	21.5330
A_{10}	0.100	0.100	0.190	0.100	0.100	0.100	0.1000	0.100	0.100	0.1000
Weight (lb)[a]	5057.88	5024.21	5529.50	5061.00	5060.92	5056.56	5060.82	5060.880	5061.42	5060.96
Average weight (lb)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	5061.95	5062.08
Standard dev. (lb)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.71	0.79
Constraint violation (%)	0.091	1.95	None	None	None	0.099	None	None	None	None
N. struct. analyses	20,000	N/A	150,000	150000	125,000	10650	1350	500×10^3	7081	16872

[a]1 lb = 0.4536 kg

better designs than those of the HS [13], PSO [25] and HPSACO [31] since the lighter designs obtained by classical HS [13], PSO [25] and HPSACO [31] violated the design constraints while the designs obtained by the SAHS [23] and TLBO [42]

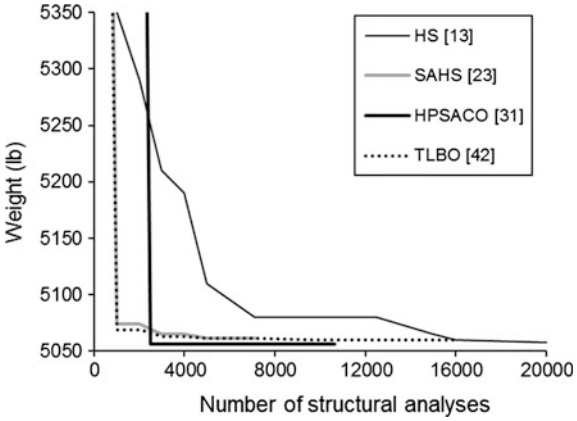

Fig. 2 Convergence histories for the ten-bar plane truss

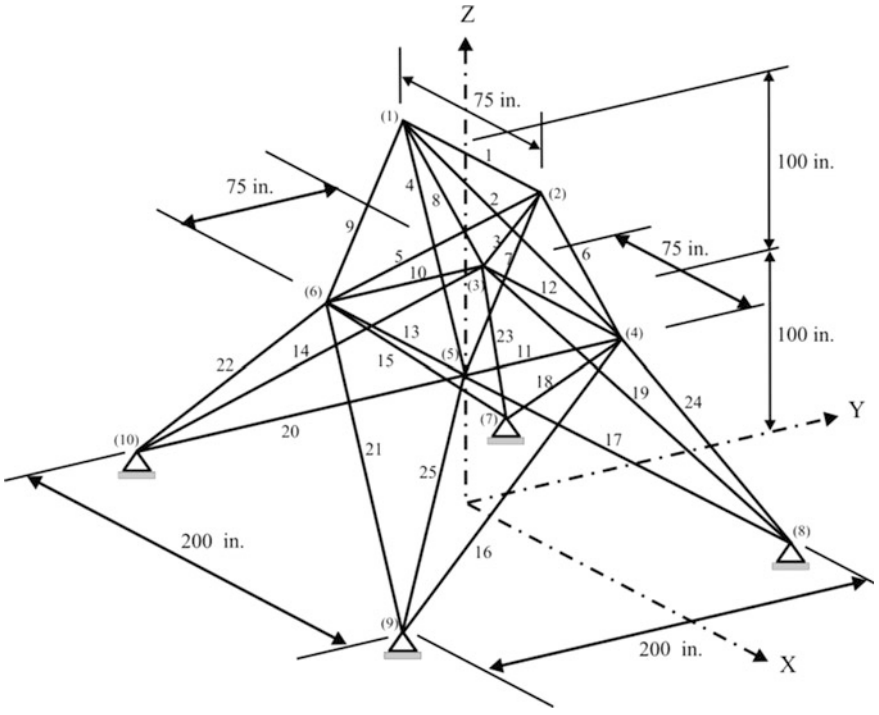

Fig. 3 Twenty-five-bar space truss

are feasible. Although IHS [22] required the lowest number of structural analyses, it should be noted that the number of structural analyses required by IHS [22] is not a significant basis of comparison to evaluate the efficiency of SAHS [23] and TLBO [42] since the IHS [22] included gradient information that allowed the number of function evaluations to be drastically reduced as gradient information, inherently speed up the optimization process [23]. Convergence histories (i.e. structural weight versus number of structural analyses) are illustrated in Fig. 2.

5.2 *Twenty-Five-Bar Space Truss*

The twenty-five-bar space truss, shown in Fig. 3 is one of the most popular design examples used in the literature for comparing different optimization methods. The Young's modulus and the density of truss members are 10^4 ksi and 0.1 lb/in^3, respectively. The structure is subject to the two loading conditions given in Table 2. The design variables of the structure and the allowable stress values for all groups are listed in Table 3. The displacement of nodes in all directions is restricted to be less than ±0.35 in. The minimum cross-sectional area for each group of elements is 0.01 in^2.

Table 2 Loading conditions for the twenty-five-bar space truss

Node	Condition 1			Condition 2		
	F_x	F_y	F_z	F_x	F_y	F_z
1	0.0	20.0	−5.0	1.0	10.0	−5.0
2	0.0	−20.0	−5.0	0.0	10.0	−5.0
3	0.0	0.0	0.0	0.5	0.0	0.0
6	0.0	0.0	0.0	0.5	0.0	0.0

Note Loads are in kips

Table 3 Allowable stress values for the twenty-five-bar space truss

Design variables A_t (in.^{2})	Allowable compressive stress (ksi)	Allowable tension stress (ksi)
A_1	35.092	40.0
A_2–A_5	11.590	40.0
A_6–A_9	17.305	40.0
A_{10}–A_{11}	35.092	40.0
A_{12}–A_{13}	35.092	40.0
A_{14}–A_{17}	6.7590	40.0
A_{18}–A_{21}	6.9590	40.0
A_{22}–A_{25}	11.082	40.0

Table 4 Optimization results obtained for the twenty-five-bar space truss

A_i (in.^{2})	[13]	[28]			[34]	[31]	[33]	[22]	Proposed [23, 42]	
	HS	PSO	PSOPC	HPSO	BB-BC	HPSACO	HBB-BC	IHS	SAHS	TLBO
A_1	0.047	9.863	0.010	0.010	0.010	0.010	2.6622	0.0100	0.010	0.0100
A_2–A_5	2.022	1.798	1.979	1.970	2.092	2.054	1.993	1.9871	2.074	2.0712
A_6–A_9	2.950	3.654	3.011	3.016	2.964	3.008	3.056	2.9935	2.961	2.9570
A_{10}–A_{11}	0.010	0.100	0.100	0.010	0.010	0.010	0.010	0.0100	0.010	0.0100
A_{12}–A_{13}	0.014	0.100	0.100	0.010	0.010	0.010	0.010	0.0100	0.010	0.0100
A_{14}–A_{17}	0.688	0.596	0.657	0.694	0.689	0.679	0.665	0.6839	0.691	0.6891
A_{18}–A_{21}	1.657	1.659	1.678	1.681	1.601	1.611	1.642	1.6769	1.617	1.6209
A_{22}–A_{25}	2.663	2.612	2.693	2.643	2.686	2.678	2.679	2.6622	2.674	2.6768
Weight (lb)	544.38	627.08	545.27	545.19	545.38	544.99	545.16	545.15	545.12 [545.38][a]	545.09 [545.38][b]
Average weight (lb)	N/A	N/A	N/A	N/A	545.78	545.52	545.66	N/A	545.94	545.41
Std. dev. (lb)	N/A	N/A	N/A	N/A	0.491	0.315	0.367	N/A	0.91	0.42
Constraint tolerance (%)	0.206	None	None	None	None	3.52	2.06	None	None	None
Num. struct. analyses	15,000	150,000	150,000	125,000	20,566	9875	12,500	1050	9051	15,318

[a]SAHS obtained a weight of 545.38 lb after 6941 analyses (the result of BB-BC)
[b]TLBO obtained a weight of 545.38 lb after 6665 analyses (the result of BB-BC)

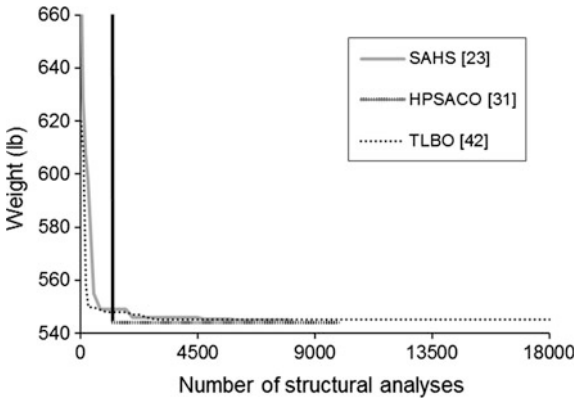

Fig. 4 Convergence history for the twenty-five-bar space truss

The results obtained by the SAHS [23], TLBO [42] and the other optimization methods existing in the literature are reported in Table 4. It is clear from Table 4 that the TLBO [42] developed the best design overall since the lighter design obtained by HS [13] and HPSACO [31] violated the design constraints while the design obtained by the TLBO [42] is feasible. Moreover, it should be noted that although the TLBO [42] required more structural analyses than the SAHS [23] to find the optimum design, the TLBO [42] developed a design with a weight of 545.38 lb after 6665 structural analyses while the SAHS [23] required 6941 structural analyses to find the same weight. The convergence histories are illustrated in Fig. 4.

5.3 Seventy-Two-Bar Space Truss

The third example deals with the design of the seventy-two-bar space truss shown in Fig. 5. The structure is subject to the loading conditions given in Table 5. The Young's modulus and density of the material are 10^4 ksi and 0.1 lb/in^3, respectively. The member cross-sectional areas are treated as design variables, and are divided into 16 groups. The allowable stress for all members is specified as 25 ksi both in tension and compression. The maximum displacements of all free nodes are limited to ±0.25 in. The minimum cross-sectional areas are specified as 0.1 in^2.

Table 6 shows the results obtained by the SAHS [23], TLBO [42] and the other optimization methods reported in current literature [13, 25, 28, 33, 34, 37]. The TLBO [42] founded the lightest design overall because the lighter design obtained by the HS [13] violates the design constraints. It is seen from Table 6 that the BB-BC [34] found a minimum weight of 379.85 lb after 19621 structural analyses for case 1 while the TLBO [42] developed the same weight after 8422 structural analyses. Figure 6 shows the convergence histories of the SAHS [23] and TLBO [42].

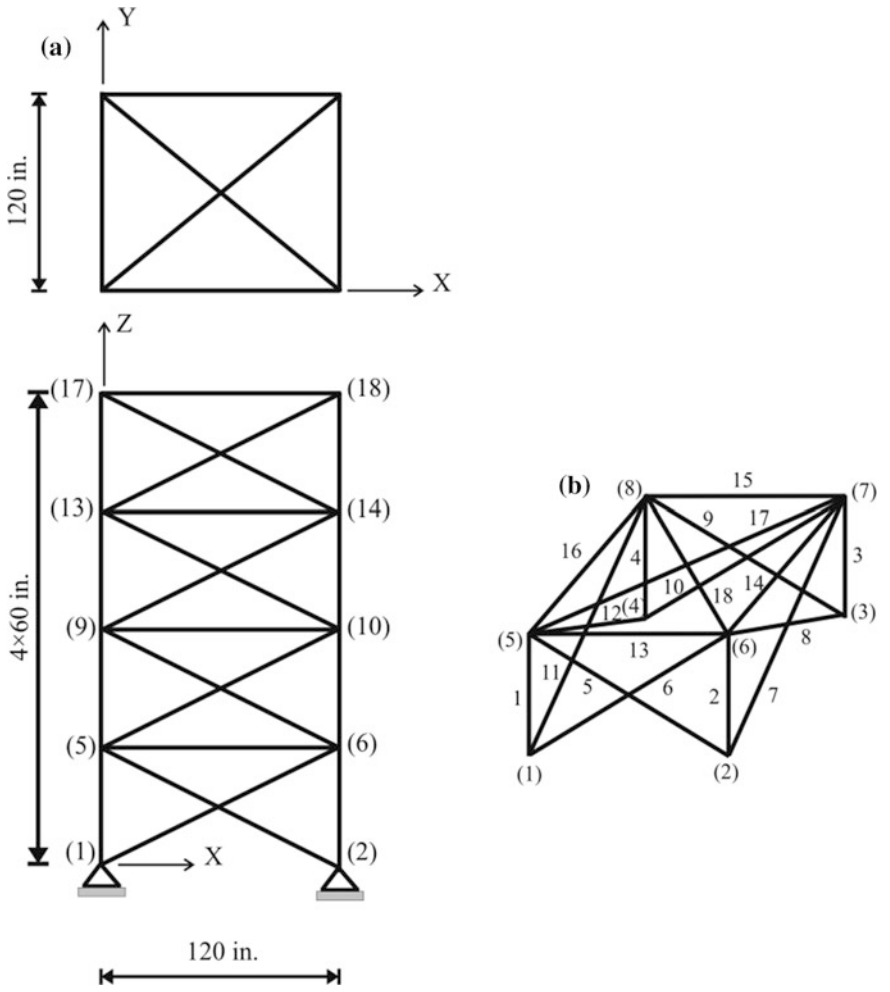

Fig. 5 Seventy-two-bar space truss: **a** dimensions **b** element and node numbering patterns for the first storey

Table 5 Loading conditions for the seventy-two-bar space truss

Node	Condition 1			Condition 2		
	F_x	F_y	F_z	F_x	F_y	F_z
17	5.0	5.0	−5.0	0.0	0.0	−5.0
18	0.0	0.0	0.0	0.0	0.0	−5.0
19	0.0	0.0	0.0	0.0	0.0	−5.0
20	0.0	0.0	0.0	0.0	0.0	−5.0

Note Loads are in kips

Table 6 Optimization results obtained for the seventy-two-bar space truss

Design variables A_i (in.^{2})	[13]	[25]	[28]			[34]	[33]	Proposed [23, 42]	
	HS	PSO	PSO	PSOPC	HPSO	BB-BC	HBB-BC	SAHS	TLBO
A_1–A_4	1.790	1.7427	41.794	1.855	1.857	1.8577	1.9042	1.860	1.9064
A_5–A_{12}	0.521	0.5185	0.195	0.504	0.505	0.5059	0.5162	0.521	0.50612
A_{13}–A_{16}	0.100	0.1000	10.797	0.100	0.100	0.1000	0.1000	0.100	0.100
A_{17}–A_{18}	0.100	0.1000	6.861	0.100	0.100	0.1000	0.1000	0.100	0.100
A_{19}–A_{22}	1.229	1.3079	0.438	1.253	1.255	1.2476	1.2582	1.271	1.2617
A_{23}–A_{30}	0.522	0.5193	0.286	0.505	0.503	0.5269	0.5035	0.509	0.5111
A_{31}–A_{34}	0.100	0.1000	18.309	0.100	0.100	0.1000	0.1000	0.100	0.100
A_{35}–A_{36}	0.100	0.1000	1.220	0.100	0.100	0.1012	0.1000	0.100	0.100
A_{37}–A_{40}	0.517	0.5142	5.933	0.497	0.496	0.5209	0.5178	0.485	0.5317
A_{41}–A_{48}	0.504	0.5464	19.545	0.508	0.506	0.5172	0.5214	0.501	0.51591
A_{49}–A_{52}	0.100	0.1000	0.159	0.100	0.100	0.1004	0.1000	0.100	0.100
A_{53}–A_{54}	0.101	0.1095	0.151	0.100	0.100	0.1005	0.1007	0.100	0.100
A_{55}–A_{58}	0.156	0.1615	10.127	0.100	0.100	0.1565	0.1566	0.168	0.1562
A_{59}–A_{66}	0.547	0.5092	7.320	0.525	0.524	0.5507	0.5421	0.584	0.54927
A_{67}–A_{70}	0.442	0.4967	3.812	0.394	0.400	0.3922	0.4132	0.433	0.40966
A_{71}–A_{72}	0.590	0.5619	18.196	0.535	0.534	0.5922	0.5756	0.520	0.56976
Weight (lb)	379.27	381.91	6818.67	369.65	369.65	379.85	379.66	380.62	379.63
									380.62
Average weight	N/A	N/A	N/A	N/A	N/A	382.08	381.85	382.42	380.20
Std. dev.	N/A	N/A	N/A	N/A	N/A	1.912	1.201	1.38	0.41
Constraint tolerance (%)	0.218	None	None	39.051	39.075	None	None	None	None
Num. struct. analyses	20,000	N/A	150,000	125,000	125,000	19,621	13,200	13,742	19,709

[a]TLBO obtained a weight of 379.85 lb after 8422 analyses (the result of BB-BC)

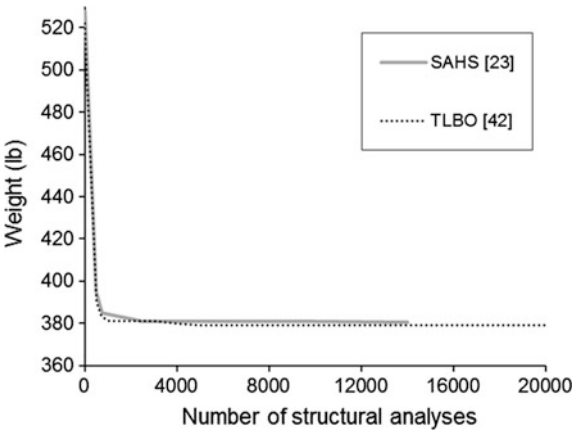

Fig. 6 Convergence histories for the seventy-two-bar space truss

6 Conclusions

The SAHS and TLBO obtained results as good as or better than other metaheuristic optimization methods in terms of both the optimum solutions and the convergence capability. It appeared that although the TLBO developed slightly heavier designs than the PSOPC, HPSO and ABC-AP in a few cases, it required significantly less structural analyses than the PSOPC, HPSO and ABC-AP in all design examples. It should be noted that standard deviation of optimized weights obtained over 20 independent runs was quite small, which is <1.0 % in all design examples, compared with average optimized weight. This points out that the SAHS and TLBO are able to find a nearly global optimum design.

References

1. Lamberti, L., Pappalettere, C.: Metaheuristic design optimization of skeletal structures: a review. Comput. Technol. Rev. **4**, 1–32 (2011)
2. Saka, M.P.: Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review. In: Topping, B.H.V. (ed.) Civil engineering computations: tools and techniques, pp. 105–147. Saxe-Coburg Publications, Stirlingshire (2007)
3. Dorigo, M., Maniezzo, V, Colorni, V.: An investigation of some properties of an ant algorithm. In: Proceedings of the 1992 Parallel Problem Solving from Nature Conference, Elsevier, Amsterdam, pp. 509–520 (1992)
4. Camp, C.V., Bichon, B.J., Stovall, S.P.: Design of steel frames ant colony optimization. J. Struct. Eng. ASCE. **131**, 369–379 (2005)
5. Camp, C.V., Bichon, B.J.: Design of space trusses using ant colony optimization. J. Struct. Eng. ASCE. **130**, 741–751 (2004)
6. Capriles, P.V.S., Fonseca, L.G., Barbosa, H.J.C., Lemonge, A.C.C.: Rank-based ant colony algorithms for truss weight minimization with discrete variables. Commun. Numer. Meth. En. **23**, 553–575 (2007)

7. Serra, M., Venini, P.: On some applications of ant colony to plane truss optimization. Struct. Multidiscip. O. **32**, 499–506 (2006)
8. Hasancebi, O., Carbas, S., Dogan, E., Erdal, F., Saka, M.P.: Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. Comput. Struct. **87**, 284–302 (2009)
9. Kaveh, A., Shojaee, S.: Optimal design of skeletal structures using ant colony optimization. Int. J. Numer. Meth. Eng. **70**, 563–581 (2007)
10. Hasancebi, O., Carbas, S., Dogan, E., Erdal, F., Saka, M.P.: Comparison of nondeterministic search techniques in the optimum design of real size steel frames. Comput. Struct. **88**, 1033–1048 (2010)
11. Kaveh, A., Talatahari, S.: An improved ant colony optimization for the design of planar steel frames. Eng. Struct. **32**, 864–873 (2010)
12. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**, 60–68 (2001)
13. Lee, K.S., Geem, Z.W.: A new structural optimization method based on the harmony search algorithm. Comput. Struct. **82**, 781–798 (2004)
14. Lee, K.S., Geem, Z.W., Lee, S.H., Bae, K.W.: The harmony search heuristic algorithm for discrete structural optimization. Eng. Optimiz. **37**, 663–684 (2005)
15. Saka, M.P.: Optimum geometry design of geodesic domes using harmony search algorithm. Adv. Struct. Eng. **10**, 595–606 (2007)
16. Saka, M.P., Erdal, F.: Harmony search based algorithm for the optimum design of grillage systems to AISC-LRFD. Struct. Multidiscip. O. **38**, 25–41 (2009)
17. Degertekin, S.O.: Harmony search algorithm for optimum design of steel frame structures: a comparative study with other optimization methods. Struct. Eng. Mech. **29**, 391–410 (2008)
18. Degertekin, S.O.: Optimum design of steel frames using harmony search algorithm. Struct. Multidiscip. O. **36**, 393–401 (2008)
19. Saka, M.P.: Optimum design of steel sway frames to BS 5950 using harmony search algorithm. J. Constr. Steel Res. **65**, 36–43 (2009)
20. Saka, M.P., Hasancebi, O.: Adaptive harmony search algorithm for design code optimization of steel structures. Harmony search algorithms for structural design optimization. In: Geem Z. W. (ed.) Studies in computational intelligence 239, Springer-Verlag, Berlin Heidelberg, pp. 79–120 (2009)
21. Hasancebi, O., Erdal, F., Saka, M.P.: Adaptive harmony search method for structural optimization. J. Struct. Eng. ASCE. **136**, 419–431 (2010)
22. Lamberti, L., Pappalettere, C.: An improved harmony-search algorithm for truss structure optimization. In: Topping B.H.V., Neves L.F.C., Barros, R.C. (eds.) Proceedings of the Twelfth International Conference Civil, Structural and Environmental Engineering Computing Civil-Comp Press, Stirlingshire, Scotland (2009)
23. Degertekin, S.O.: An improved harmony search algorithms for sizing optimization of truss structures. Comput. Struct. **92–93**, 229–241 (2012)
24. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
25. Perez, R.E., Behdinan, K.: Particle swarm approach for structural design optimization. Comput. Struct. **85**, 1579–1588 (2007)
26. Fourie, P., Groenwold, A.: The particle swarm optimization algorithm in size and shape optimization. Struct. Multidiscip. O. **23**, 259–267 (2002)
27. Schutte, J.F., Groenwold, A.: Sizing design of truss structures using particle swarms. Struct. Multidiscip. O. **25**, 261–269 (2009)
28. Li, L.J., Huang, Z.B., Liu, F., Wu, Q.H.: A heuristic particle swarm optimizer for optimization of pin connected structures. Comput. Struct. **85**, 340–349 (2007)
29. Li, L.J., Huang, Z.B., Liu, F.: A heuristic particle swarm optimization method for truss structures with discrete variables. Comput. Struct. **87**, 435–443 (2009)
30. Kaveh, A., Talathari, S.: A particle swarm ant colony optimization for truss structures with discrete variables. J. Constr. Steel Res. **65**, 1558–1568 (2009)

31. Kaveh, A., Talatahari, S.: Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput. Struct. **87**, 267–283 (2009)
32. Erol, O.K., Eksin, I.: New optimization method: big bang-big crunch. Adv. Eng. Softw. **37**, 106–111 (2006)
33. Kaveh, A., Talatahari, S.: Size optimization of space trusses using big-bang big-crunch algorithm. Comput. Struct. **87**, 1129–1140 (2009)
34. Camp, C.V.: Design of space trusses using big bang-big crunch optimization. J. Struct. Eng. ASCE. **133**, 999–1008 (2007)
35. Kaveh, A., Talatahari, S.: Optimal design of Schwedler and ribbed domes via hybrid big bang-big crunch algorithm. J. Constr. Steel Res. **66**, 412–419 (2010)
36. Karaboga, D.: An idea based on honey bee swarm for numerical optimization, technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
37. Sonmez, M.: Artificial bee colony algorithm for optimization of truss optimization. Appl. Soft Comput. **11**, 2406–2418 (2011)
38. Sonmez, M.: Discrete optimum design of truss structures using artificial bee colony algorithm. Struct. Multidiscip. O. **43**, 85–97 (2011)
39. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comp. Aid. Des. **43**, 303–315 (2011)
40. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf. Sci. **183**, 1–15 (2012)
41. Toğan, V.: Design of steel frames using teaching-learning based optimization. Eng. Struct. **34**, 225–232 (2012)
42. Degertekin, S.O., Hayalioglu, M.S.: Sizing optimization of truss structures using teaching-learning-based optimization. Comput. Struct. **119**, 177–188 (2013)

Performance-Based Optimum Seismic Design of Steel Dual Braced Frames by Bat Algorithm

Saeed Gholizadeh and Hamed Poorhoseini

Abstract One of the challenging problems in the field of structural engineering is designing cost-efficient structures with improved performance subject to earthquake loading conditions. Structural optimization procedures can be effectively employed for performance-based optimal design of structures. In this study, bat algorithm (BA) is utilized to implement performance-based optimum seismic design of steel dual braced frames for various performance levels. The required structural seismic responses are evaluated by performing nonlinear pushover analysis. The results found by BA are then compared with those of obtained by other popular meta-heuristics such as firefly algorithm (FA) and particle swarm optimization (PSO) to provide an insight about its computational performance. Two numerical examples are presented and the numerical results reveal that the BA outperforms PSO and FA.

Keywords Meta-heuristic · Bat algorithm · Optimization · Performance-based design · Earthquake · Steel structure · Dual braced frame

1 Introduction

One of the most important issues in designing a structural system is its sufficient seismic resistance to ensure availability after an earthquake. In recent years, the concepts of performance-based design were developed and applied in the framework of powerful and reliable seismic design procedures. On the other hand, the seismic performance of structural systems can be affected by a large number of parameters and therefore, recognizing that the current design is the best solution or still there is room for finding cost-efficient solutions is a quite difficult task. In the face of increase in price of materials, finding cost-efficient structural designs, with

S. Gholizadeh (✉) · H. Poorhoseini
Department of Civil Engineering, Urmia University, Urmia, Iran
e-mail: s.gholizadeh@urmia.ac.ir

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*,
Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_5

improved performance, is one of the major concerns in the field of structural engineering. In order to achieve this purpose, structural optimization methodologies have been developed during the last decades. The performance-based design of steel structures in the framework of structural optimization is a topic of growing interest [1–4]. In the performance-based design approaches, nonlinear analysis procedures are efficiently employed to evaluate the seismic response of structures.

Performance levels are usually divided into three categories: immediate occupancy (IO), life safety (LS), and collapse prevention (CP). The IO level implies very light damage with minor local yielding and negligible residual drifts. In the LS level, the structure tolerates sever damage, but it remains safe for the occupants to evacuate the building. The CP level is associated with extensive inelastic distortion of structural members and an increase in load or deflection results in collapse of the structure. The performance-based design methods tend to consider the nonlinear seismic response of structures. These methods directly address inelastic deformations to identify the levels of damage during severe seismic events. A nonlinear analysis tool is required to evaluate earthquake demands at the various performance levels. Pushover analysis is widely adopted as the effective tool for such nonlinear analysis because of its simplicity compared with dynamic nonlinear procedures. The purpose of the nonlinear static pushover analysis is to assess structural performance in terms of strength and deformation capacity globally as well as at the element level during the incremental loading process [2].

The traditional performance-based design process can be effectively replaced by an automatic advanced procedure for structural seismic design utilizing structural optimization algorithms. In the last years, many researches have been done in the field of performance-based optimum design of structures. However, a few researches have utilized meta-heuristics. During the last decades, meta-heuristic methods have received considerable attention and have experienced rapid development. Their popularity lies in their ease of implementation and their ability to find global or near global optimum designs. Their ability to explore design space makes them more suitable for handling complex optimization problems with highly nonlinear objective functions with many local optima. A number of works which address performance-based optimum design employing meta-heuristics are reviewed here. Liu et al. [5] proposed a genetic algorithm (GA) based multi-objective structural optimization procedure for steel frames considering weight, maximum inter-story drift for two-performance levels, and design complexity criteria objective functions. Fragiadakis and Lagaros [6] presented a methodology based on evolution strategies (ES) for the performance-based optimum design of steel structures subjected to seismic loading considering inelastic behavior and life-cycle cost to take into account the impact of future earthquakes during the design phase. Kaveh et al. [7] compared the computational performance of ant colony optimization (ACO) and genetic algorithm (GA) meta-heuristics for performance-based optimal seismic design of frame structures. Gholizadeh and Moghadas [2] proposed an improved quantum particle swarm optimization (IQPSO) meta-heuristic algorithm for tackling the problem of performance-based optimum design of steel frames. Gholizadeh

[1] proposed a computational methodology for performance-based optimum seismic design of steel moment frames including a modified firefly algorithm (MFA) for performing optimization and a wavelet cascade-forward back-propagation (WCFBP) new neural network for prediction the results of nonlinear pushover analysis during the optimization process.

One of the efficient meta-heuristics which was proposed by Yang [8] is bat algorithm (BA). The BA is based on the echolocation behavior of bats. A number of successful applications of BA in the field of structural engineering have been reported in literature [9–11]. The main aim of this study is to implement performance-based optimum seismic design of dual steel frame together with X-bracing utilizing BA. The optimization task is also achieved by firefly algorithm (FA) and particle swarm optimization (PSO) for comparing the results. Two illustrative examples and the numerical results reveal that BA possesses better computational performance compared with PSO and FA.

2 Performance-Based Optimum Seismic Design

According to current design codes the strength of the structure is evaluated at one limit-state and the serviceability is usually checked in order to ensure that the structure will not deflect excessively. Performance-based seismic design (PBSD) methodology differs from seismic design procedures for the design of new buildings specified in the current building design codes. PBSD is a design procedure in which the seismic demands of a structure are determined at predefined performance levels by introducing design checks at a higher level to ensure reliable and predictable seismic performance over its life. The definition of the performance objectives is the fundamental part in PBSD. A performance objective is defined as a given level of performance for a specific hazard level. To define a performance objective, at first the level of structural performance should be selected and then the corresponding seismic hazard level should be determined. In the present chapter, IO, LS, and CP performance levels are considered according to FEMA-273 [12]. Each objective corresponds to a given probability of being exceed during 50 years. A usual assumption [4] is that the IO, LS, and CP performance levels correspond, respectively, to a 20, 10, and 2 % probability of exceedance in 50 year period. In this study, the mentioned hazard levels are considered [1].

In order to achieve PBSD, the structural nonlinear responses should be evaluated and in the present study the nonlinear static pushover analysis is conducted to quantify seismic induced nonlinear response of structures. The displacement coefficient method [13] is adopted in this work to evaluate the seismic demands on building frameworks under equivalent static earthquake loading. In this method the structure is pushed with a specific distribution of the lateral loads until the target displacement is reached. The target displacement can be obtained from the FEMA-356 [13] as follows:

$$\delta_t = C_0 C_1 C_2 C_3 S_a \frac{T_e^2}{4\pi^2} g \tag{1}$$

where C_0 relates the spectral displacement to the likely building roof displacement; C_1 relates the expected maximum inelastic displacements to the displacements calculated for linear elastic response; C_2 represents the effect of the hysteresis shape on the maximum displacement response; and C_3 accounts for P-Δ effects. The T_e is the effective fundamental period of the building in the direction under consideration; S_a is the response spectrum acceleration corresponding to the T_e.

In this study, the Open Sees [14] platform is employed to perform the pushover analysis. In order to ensure that the obtained designs possess desirable performance, according to the employed design code, several constraints should be considered during the design process. These constraints can be checked in two steps as follows:

2.1 Serviceability Checks

The structure is checked for gravity loads. To perform serviceability checks, the following load combinations (Q_G^{SC}) are taken into account [1]:

$$Q_G^{\mathrm{SC}} = \begin{cases} Q_D \\ Q_D + Q_L \\ 1.4Q_D \\ 1.2Q_D + 1.6Q_L \end{cases} \tag{2}$$

where Q_D and Q_L are dead and live loads, respectively.

In this work, the checks of LRFD-AISC [15] code must be satisfied as follows for the non-seismic load combinations for all structural elements:

$$\text{for } \frac{P_u}{\varphi_c P_n} < 0.2;\ g_{\sigma,l}(X) = \left[\frac{P_u}{2\varphi_c P_n} + \left(\frac{M_{ux}}{\varphi_b M_{nx}} + \frac{M_{uy}}{\varphi_b M_{ny}}\right)\right] - 1 \le 0, \quad l = 1, \ldots, \mathrm{ne} \tag{3}$$

$$\text{for } \frac{P_u}{\varphi_c P_n} \ge 0.2;\ g_{\sigma,l}(X) = \left[\frac{P_u}{\varphi_c P_n} + \frac{8}{9}\left(\frac{M_{ux}}{\varphi_b M_{nx}} + \frac{M_{uy}}{\varphi_b M_{ny}}\right)\right] - 1 \le 0, \quad l = 1, \ldots, \mathrm{ne} \tag{4}$$

where P_u is the required strength (tension or compression); P_n is the nominal axial strength (tension or compression); φ_c is the resistance factor; M_{ux} and M_{uy} are the required flexural strengths in the x and y directions, respectively; M_{nx} and M_{ny} are the nominal flexural strengths in the x and y directions; and φ_b is the flexural resistance reduction factor ($\varphi_b = 0.9$); and X is the vector of design variables.

As the completion of the first step is prerequisite for the second one, if the checks of the first step are not satisfied then the candidate design is rejected, else a nonlinear pushover analysis based on the displacement coefficient method is performed in order to estimate the capacity of the structure in different performance levels. The structural capacity is associated to the maximum inter-story drift values. The constraints of the second step can be described as follows [1]:

2.2 Ultimate Limit-State Checks

The applied PBSD concept is a displacement-based design procedure where the design criteria and the capacity demand levels are expressed in terms of displacements rather than forces [16]. To perform ultimate limit-state checks, the lateral drifts should be determined at various performance levels. In pushover analysis, the lateral load distribution in the height of the frame is defined as follows [12]:

$$P_s = V_b\left(\frac{G_s H_s^k}{\sum_{m=1}^{\mathrm{ns}} G_m H_m^k}\right) \tag{5}$$

where P_s = lateral load applied at story s; V_b = base shear; H_s, H_m = height from the base of the building to stories s and m, respectively; G_s, G_m = seismic weight for story levels s and m, respectively; k = constant number determined by period and in this chapter k is chosen to be 2; ns is the number of stories.

The following component gravity forces, Q_G^{PBD}, is considered for combination with seismic loads according to [12].

$$Q_G^{\mathrm{PBD}} = 1.1(Q_D + Q_L) \tag{6}$$

In order to implement pushover analysis to evaluate the seismic demands of the structures, the target displacement should be determined. To achieve this task, S_a should be calculated for the three performance levels. In this case three acceleration design spectra, which represent three different earthquake levels corresponding to 20, 10, and 2 % probability of exceeding in a 50 year period, are taken as the basis for calculating the seismic loading for the three performance levels IO, LS, and CP, respectively. Without loss of generality, the calculation of spectral acceleration S_a^i for each design spectrum i can be expressed as:

$$S_a^i = \begin{cases} F_a S_s^i (0.4 + 3T/T_0) & \text{if} \quad 0 < T \le 0.2T_0^i \\ F_a S_s^i & \text{if} \quad 0.2T_0^i < T \le T_0^i \\ F_v S_1^i / T & \text{if} \quad T > T_0^i \end{cases}, \quad i = \mathrm{IO, LS, CP} \tag{7}$$

Table 1 Performance level site parameters for site class of D

Performance level	Hazard level	S_s (g)	S_1 (g)	F_a	F_v
IO	20 %/50 years	0.658	0.198	1.27	2.00
LS	10 %/50 years	0.794	0.237	1.18	1.92
CP	2 %/50 years	1.150	0.346	1.04	1.70

$$T_0^i = \frac{F_v S_1^i}{F_a S_s^i} \tag{8}$$

where T is the elastic fundamental period of the structure, which is computed here from structural modal analysis; S_s^i and S_1^i are the short-period and the first sec.-period response acceleration parameters, respectively. F_a and F_v are the site coefficients determined from FEMA-273 [12], based on the site class and the values of the response acceleration parameters S_s^i and S_1^i, respectively, according to Table 1 [7].

The lateral drift constraints at various performance levels can be expressed as follows:

$$g_{d,j}^i = \frac{d_j^i}{d_{\text{all}}^i} - 1 \le 0, \quad j = 1, 2, \ldots, \text{ns}, \quad i = \text{IO, LS, CP} \tag{9}$$

where d_j^i is the jth story drift of a steel dual braced frame associated with ith performance level; d_{all}^i is the allowable values which in this study are chosen to be 0.5, 1.5, and 2.0 %, for IO, LS, and CP performance levels, respectively [12].

The plastic rotation constraints for beams and columns at various performance levels are as follows:

$$g_{\theta,l}^i = \frac{\theta_l^i}{\theta_{\text{all}}^i} - 1 \le 0, \quad l = 1, 2, \ldots, \text{ne}, \quad i = \text{IO, LS, CP} \tag{10}$$

where θ_k^i and $\theta_{\text{all}}^{\text{i}}$ are the kth element plastic rotation and its allowable value associated with ith performance level.

In FEMA-356 [13] θ_{all} for IO, LS, and CP performance levels is chosen to be θ_y, $6\theta_y$, and $8\theta_y$, respectively. For beams and columns θ_y can be computed as:

$$\theta_y = \begin{cases} \frac{\text{ZF}_{ye} l_b}{6\text{EI}_b} & \text{Beams} \\ \frac{\text{ZF}_{ye} l_c}{6\text{EI}_c}\left(1 - \frac{P}{P_{ye}}\right) & \text{Columns} \end{cases} \tag{11}$$

where Z is plastic section modulus, F_{ye} is expected yield strength of the material, E is modulus of elasticity, l_b and I_b are length and moment of inertia of a beam, respectively, l_c and I_c are length and moment of inertia of a column, respectively, P is axial force in the column at the target displacement and P_{ye} is expected axial yield force of the column.

The axial deformation constraints for braces at various performance levels are as follows:

$$g^i_{\Delta,k} = \frac{\Delta^i_k}{\Delta^i_{\text{all}}} - 1 \leq 0, \quad k = 1, 2, \ldots, \text{nb}, \quad i = \text{IO}, \text{LS}, \text{CP} \tag{12}$$

where Δ^i_l is the axial deformation of a brace at ith performance level. Δ^i_{all} for braces in compression at IO, LS, and CP performance levels is chosen to be $0.25\Delta_C$, $5\Delta_C$, and $7\Delta_C$, respectively, in which Δ_C is the axial deformation at expected buckling load. Δ^i_{all} for braces in tension at IO, LS, and CP performance levels is chosen to be $0.25\Delta_T$, $7\Delta_T$, and $9\Delta_T$, respectively in which Δ_T is the axial deformation at expected tensile yielding load. *nb* is the number of braces.

In this study, for modeling nonlinear behavior of beams and columns a simple bilinear stress–strain relationship with 3 % kinematic hardening is considered. For modeling braces, uniaxial co-rotational truss element is used. As shown in Fig. 1 the hardening rule is bilinear kinematics in tension. In compression, according to FEMA274 [17], it is assumed that the element buckles at its corresponding buckling stress state and its residual stress is about 20 % of the buckling stress.

In this figure, σ_{cr} and σ_y are buckling and yield stresses, respectively and ε_{cr} and ε_y are their corresponding strains. In this model, the buckling stress of braces is computed based on the AISC-LRFD code [15] as follows:

$$\sigma_b = \begin{cases} (0.658^{\lambda_c^2})\sigma_y & \lambda_c \leq 1.5 \\ \left(\frac{0.877}{\lambda_c^2}\right)\sigma_y & \lambda_c > 1.5 \end{cases}, \quad \lambda_c = \frac{\text{KL}}{r\pi}\sqrt{\frac{\sigma_y}{E}} \tag{13}$$

where λ_c is slenderness parameter; K is effective length factor of braces.

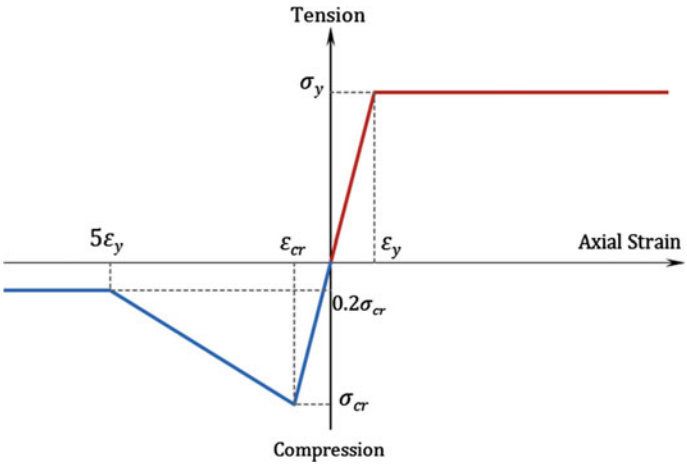

Fig. 1 Stress–strain relationship for braces

2.3 Optimization Problem Statement

The main aim of a sizing structural optimization problem is usually to minimize the weight of the structure subject to a number of design constraints. For a steel structure with *ne* members that are collected in *ng* design groups, if the variables associated with each design group are selected from a given profile list of steel sections (such as W-shaped sections), a discrete optimization problem can be formulated as follows:

$$\text{Find: } X = \left\{x_1\, x_2 \cdots x_i \cdots x_{\text{ng}}\right\}^{\text{T}} \tag{14}$$

$$\text{To minimize: } w(X) = \sum_{i=1}^{\text{ng}} \rho_i A_i \sum_{j=1}^{\text{nm}} L_j \tag{15}$$

$$\text{Subject to: } g_k(X) \le 0, \quad k = 1, 2, \ldots, \text{ nc} \tag{16}$$

where x_i is an integer value expressing the sequence numbers of steel sections assigned to *i*th group; *w* represents the weight of the frame, ρ_i and A_i are weight of unit volume and cross-sectional area of the *i*th group section, respectively; nm is the number of elements collected in the *i*th group; L_j is the length of the *j*th element in the *i*th group; $g_k(X)$ is the *k*th behavioral constraint and *nc* is the number of constraints.

In this study, the constraints of performance-based optimum seismic design (PBOSD) problem is handled using the concept of exterior penalty functions method (EPFM) [18]. The general approach of penalty function methods is to minimize the objective function as an unconstrained function but to provide some penalty to limit constraint violations. In this case, the pseudo-unconstrained objective function is expressed as follows:

$$\Phi(X, r_p) = w(X)(1 + \text{PF}_\sigma + \text{PF}_d + \text{PF}_\theta + \text{PF}_\Delta) \tag{17}$$

$$\text{PF}_\sigma = r_p \sum_{l=1}^{\text{ne}} \left(\max\{0, g_{\sigma,l}\}\right)^2 \tag{18}$$

$$\text{PF}_d = r_p \sum_{i}^{\text{IO,LS,CP}} \sum_{j=1}^{\text{ns}} \left(\max\{0, g^i_{d,j}\}\right)^2 \tag{19}$$

$$\text{PF}_\theta = r_p \sum_{i}^{\text{IO,LS,CP}} \sum_{l=1}^{\text{ne}} \left(\max\{0, g^i_{\theta,l}\}\right)^2 \tag{20}$$

$$\text{PF}_\Delta = r_p \sum_{i}^{\text{IO,LS,CP}} \sum_{k=1}^{\text{nb}} \left(\max\{0, g^i_{\Delta,k}\}\right)^2 \tag{21}$$

where Φ, PF_{σ}, PF_{d}, PF_{θ}, PF_{Δ}, and r_p are the pseudo objective function, penalty function for serviceability check, penalty function for drift checks, penalty function for element plastic rotation checks, penalty function for braces' axial deformation checks, and positive penalty parameter, respectively.

3 Meta-Heuristic Algorithms

Meta-heuristics are popular due to their interesting characteristics for solving a wide range of continuous, discrete, and combinatorial complex optimization problems. Most of the meta-heuristics mimic natural phenomena to search the design space of the optimization problems. In recent years, it was found that meta-heuristic algorithms are computationally efficient even if greater number of optimization cycles is needed to reach the optimum in comparison with gradient-based algorithms. Furthermore, meta-heuristic algorithms are more robust in finding the global optima, due to their random search, whereas mathematical programming algorithms may be trapped into local optima. In this work, PSO, FA, and BA meta-heuristic algorithms are employed to tackle the stated optimization problem. The main concepts of the mentioned meta-heuristics are explained as follows:

3.1 Particle Swarm Optimization

PSO is a population-based meta-heuristic algorithm in which the population is called swarm and each individual in the swarm is called particle. Each particle of the swarm represents a potential solution of the optimization problem. This meta-heuristic algorithm first presented by Eberhart and Kennedy [19]. The particles fly through the search space and their positions are updated based on the best positions of individual particles in each iteration. Using following equations the position of the particles is updated:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \left(P_{\text{best},i}^k - X_i^k \right) + c_2 r_2 \left(G_{\text{best}}^k - X_i^k \right) \tag{22}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{23}$$

where X_i and V_i represent the current position and the velocity of the ith particle respectively; $P_{\text{best},i}^k$ is the best previous position of the ith particle (called $p$best) and G_{best}^k is the best global position among all the particle in the swarm (called $g$best); Positive constants c_1 and c_2 are the cognitive and social components, respectively; r_1 and r_2 are two uniform random sequences generated from range (0, 1) and ω is the inertia weight used discount the previous velocity of the particle persevered.

Due to the importance of ω in achieving an efficient search behavior, the updating criterion can be taken as follows:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{k_{\max}} \cdot k \tag{24}$$

where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum values of ω, respectively. Also, $k_{\max}$, and k are the number of maximum iterations and the number of present iteration, respectively.

In this work, the internal parameters of PSO are as follows: $c_1 = 2.0$, $c_2 = 2.0$, $\omega_{\max} = 0.5$ and $\omega_{\min} = 0.0$.

3.2 *Firefly Algorithm*

The FA, introduced by Yang [20], is a meta-heuristic optimization algorithm inspired by the flashing behavior of fireflies. FA is a population-based algorithm, which may share many similarities with PSO. Fireflies communicate, search for pray, and find mates using bioluminescence with varied flashing patterns [21]. In order to develop the firefly algorithm, natural flashing characteristics of fireflies have been idealized using the following three rules [20]:

a. All of the fireflies are unisex; therefore, one firefly will be attracted to other fireflies regardless of their sex.
b. Attractiveness of each firefly is proportional to its brightness, thus for any two flashing fireflies, the less bright firefly will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
c. The brightness of a firefly is determined according to the nature of the objective function.

The attractiveness of a firefly is determined by its brightness or light intensity which is obtained from the objective function of the optimization problem. However, the attractiveness β, which is related to the judgment of the beholder, varies with the distance between two fireflies. The attractiveness β can be defined by [22]

$$\beta = \beta_0 e^{-\gamma . r^2} \tag{25}$$

where r is the distance of two fireflies, β_0 is the attractiveness at $r = 0$, and γ is the light absorption coefficient.

The distance between two fireflies i and j at X_i and X_j, respectively, is determined using the following equation:

$$r_{ij} = \left\| X_i - X_j \right\| = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k} \right)^2} \tag{26}$$

where $x_{i,k}$ is the kth parameter of the spatial coordinate x_i of the ith firefly.

In the firefly algorithm, the movement of a firefly i towards a more attractive (brighter) firefly j is determined by the following equation [22]:

$$X_i = X_i + \beta_0 e^{-\gamma . r_{ij}^2} (X_j - X_i) + \alpha(\text{rand} - 0.5) \tag{27}$$

where the second term is related to the attraction, while the third term is randomization with α being the randomization parameter. Also rand is a random number generator uniformly distributed in [0, 1].

In this work, the modified equation proposed in [23] for computing α is employed as follows:

$$\alpha = \alpha_{\max} - \frac{\alpha_{\max} - \alpha_{\min}}{t_{\max}} \cdot t \tag{28}$$

where $\alpha_{\max}$ and $\alpha_{\min}$ are the maximum and minimum values of α. Also, $t_{\max}$ and t are the numbers of maximum iterations and present iteration, respectively.

In this work, the internal parameters of FA are as follows: $\beta_0 = 1.0$, $\gamma = 1.0$, $\alpha_{\max} = 1.0$, and $\alpha_{\min} = 0.2$.

3.3 Bat Algorithm

The BA meta-heuristic optimization algorithm is inspired from the echolocation behavior of microbats. Echolocation is an advanced hearing based navigation system used by bats to detect objects in their surroundings by emitting a sound to the environment. While they are hunting for preys or navigating, these animals produce a sound wave that travels across the canyon and eventually hits an object or a surface and returns to them as an echo. The sound waves travel at a constant speed in zones where atmospheric air pressure is identical. By following the time delay of the returning sound, these animals can determine the precise distance to circumjacent objects. Further, the relative amplitudes of the sound waves received at each individual ear are used to identify shape and direction of the objects. The information collected this way of hearing is synthesized and processed in the brain to depict a mental image of their surroundings [24].

The echolocation characteristics of microbats in BA are idealized as the following rules [25]:

1. All bats use echolocation to sense distance, and they also "know" the difference between food/prey and background barriers in some magical way;
2. Bats randomly fly with velocity V_i at position X_i with a fixed frequency $f_{\min}$, varying wavelength λ and loudness A^0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target;
3. Although the loudness can vary in many ways, it is assumed that the loudness varies from a large (positive) A^0 to a minimum constant value $A_{\min}$;

The position and velocity of each bat should be updated in the design space according to the following equations:

$$f_i = f_{\min} + (f_{\max} - f_{\min})u_i \tag{29}$$

$$V_i^{k+1} = \text{round}\left(V_i^k + \left(X_i^k - X^*\right)f_i\right) \tag{30}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{31}$$

where $f_{\min}$ and $f_{\max}$ are the lower and upper bounds imposed for the frequency range of bats. $u_i \in [0, 1]$ is a vector containing uniformly distribution random numbers; X^* is the current global best solution;

A local search is implemented on a randomly selected bat from the current population using the following equation:

$$X^{k+1} = X^k + \text{round}(\varepsilon_j A^{k+1}) \tag{32}$$

where ε_j is a uniform random number in $[-1, 1]$ selected for each design variable of the selected bat. A^{k+1} is the average loudness of all the bats at the current iteration. Also, "round" is to discrete the continuous space into an integer/discrete one.

The loudness A_i and the rate r_i of pulse emission have to be updated accordingly as the iterations proceed. As the loudness usually decreases once a bat has found its prey while the rate of pulse emission increases, the loudness can be chosen as any value of convenience. In this work, $A^0 = 1$ and $A_{\min} = 0$ also, $r^0 = 0$ and $r_{\max} = 1$.

$$A_i^{k+1} = \alpha A_i^k \tag{33}$$

$$r_i^{k+1} = r_i^0\left(1 - \mathrm{e}^{-\gamma.k}\right) \tag{34}$$

where α and γ are constants.

In this work, the internal parameters of BA are as follows: $\alpha = 0.9$, $\gamma = 0.01$, $f_{\min} = 0.0$, and $f_{\max} = 1.0$.

4 Numerical Results

Two numerical examples of steel dual braced frames are presented. It is assumed that beams and columns are rigidly connected to each other and the X-bracings are connected to the frames with pinned ends. In these examples, the modulus of elasticity is 204 GPa and yield stress for columns is 351.53 MPa and for beams and bracings is 253.1 MPa. The dead load of Q_D = 2500 kg/m and live load of Q_L = 1500 kg/m are applied to the all beams. All of the required computer programs for performing optimization tasks are coded in MATLAB [26] platform. Furthermore, for computer implementation a personal Pentium IV 3.0 GHz has been used.

4.1 Example 1: Six-Story, Five-Bay Steel Dual Braced Frame

The first example is the six-story, five-bay steel dual braced frame which is shown in Fig. 2 with elements grouping details. In this example there are eight design variables associated with four columns, two beams, and two bracing cross sections.

For performing optimization, the number of particles in the swarm is 50 and the maximum number of iterations is limited to 200.

The results of optimization are presented in Table 2 and convergence histories of the PSO, FA and BA are compared in Fig. 3.

The results of Table 2 imply that BA finds an optimal solution which is 3.62 and 1.74 % lighter than those of found by PSO and FA, respectively. It can be also observed that FA outperforms PSO.

The convergence histories presented in Fig. 3 demonstrate that the convergence rate of the BA is better compared with PSO and FA and in the mean time FA possesses better convergence behavior in comparison with PSO.

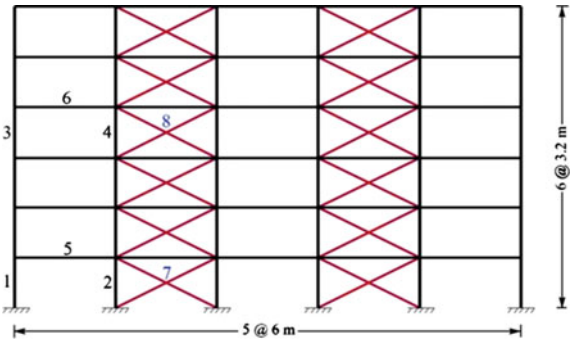

Fig. 2 The six-story steel dual braced frame with X-bracings in bays 2 and 4

Table 2 Optimum designs of six-story steel dual braced frame

Design variable No.	Optimum designs		
	PSO	FA	BA
1	W8 × 31	W16 × 45	W14 × 34
2	W40 × 149	W27 × 129	W36 × 135
3	W14 × 30	W14 × 30	W14 × 30
4	W24 × 68	W8 × 35	W16 × 36
5	W12 × 45	W12 × 45	W12 × 45
6	W21 × 62	W21 × 62	W12 × 58
7	W24 × 68	W18 × 71	W18 × 71
8	W18 × 46	W18 × 50	W18 × 50
Optimal weight (kg)	51092.64	50112.88	49241.98
Constraints violation	0.0	0.0	0.0

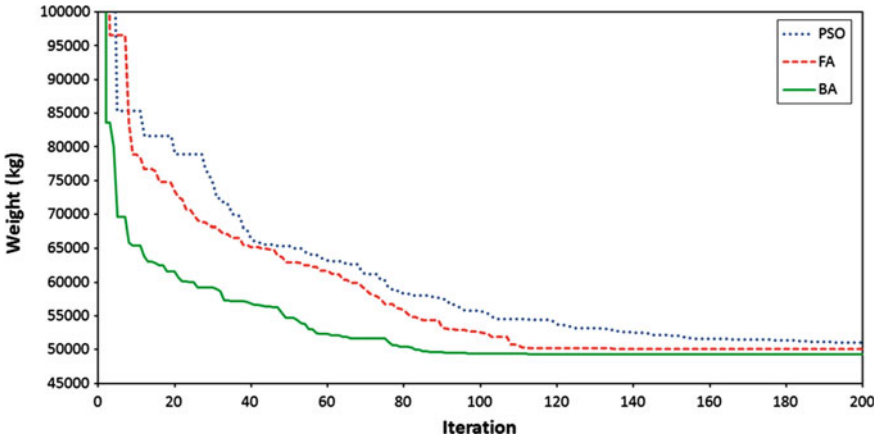

Fig. 3 Convergence histories of PSO, FA and BA for optimization of six-story steel dual braced frame

The results reveal that axial deformation constraints for braces at IO performance level are active and therefore dominate the optimal design while all of the other constraints are satisfied. Figure 4 depicts the values of Δ/Δ_{all} for all braces at IO level for the optimal design found by BA.

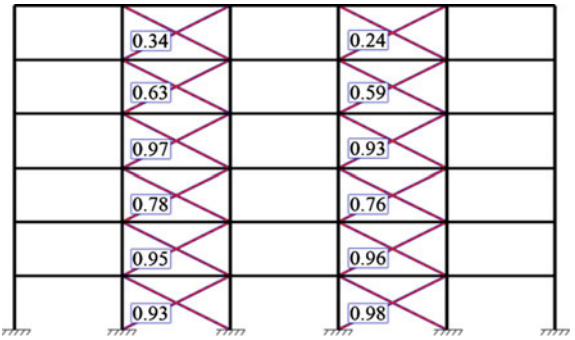

Fig. 4 Δ/Δ_{all} values at IO level for the optimal six-story steel dual braced frame found by BA

4.2 *Example 2: Twelve-Story, Five-Bay Steel Dual Braced Frame*

Figure 5 shows the twelve-story steel frame with its elements grouping details.

Design variable vector in this example includes sixteen components which are divided to eight columns, four beams, and four bracing group cross sections. The number of particles in the swarm and the maximum number of iterations are chosen to be 50 and 400, respectively.

Table 3 reports the optimal designs found by PSO, FA, and BA and their convergence histories are depicted in Fig. 6.

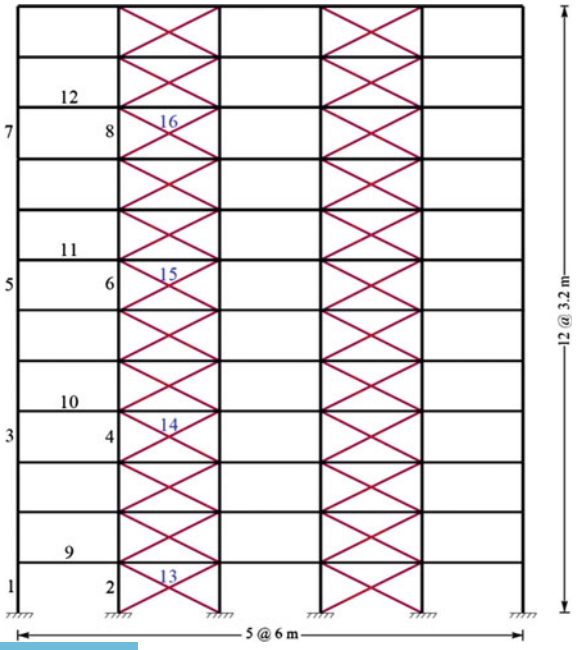

Fig. 5 The twelve-story steel dual braced frame with X-bracings in bays 2 and 4

Table 3 Optimum designs of twelve-story steel dual braced frame

Design variable No.	Optimum designs		
	PSO	FA	BA
1	W30 × 90	W30X90	W30X90
2	W36 × 160	W40 × 167	W33 × 141
3	W30 × 90	W10 × 68	W27 × 84
4	W30 × 108	W30 × 108	W27 × 84
5	W21 × 68	W12 × 45	W21 × 48
6	W24 × 76	W27 × 94	W24 × 68
7	W16 × 31	W16 × 31	W16 × 40
8	W24 × 68	W30 × 90	W21 × 48
9	W18 × 97	W8 × 67	W18 × 55
10	W8 × 67	W12 × 45	W16 × 89
11	W10 × 39	W18 × 50	W18 × 50
12	W16 × 40	W16 × 40	W18 × 50
13	W10 × 60	W10 × 60	W21 × 68
14	W16 × 50	W21 × 57	W21 × 57
15	W18 × 46	W18 × 46	W18 × 46
16	W6 × 25	W10 × 19	W6 × 25
Optimal weight (kg)	84018.92	82544.74	81274.68
Constraints violation	0.0	0.0	0.0

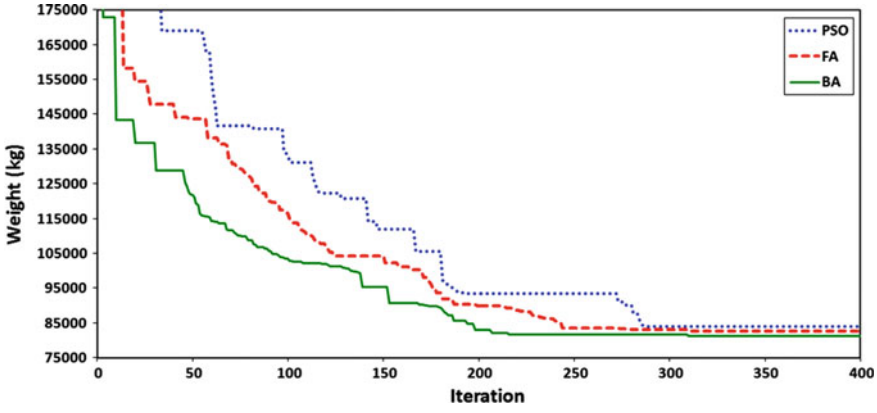

Fig. 6 Convergence histories of PSO, FA, and BA for optimization of twelve-story steel dual braced frame

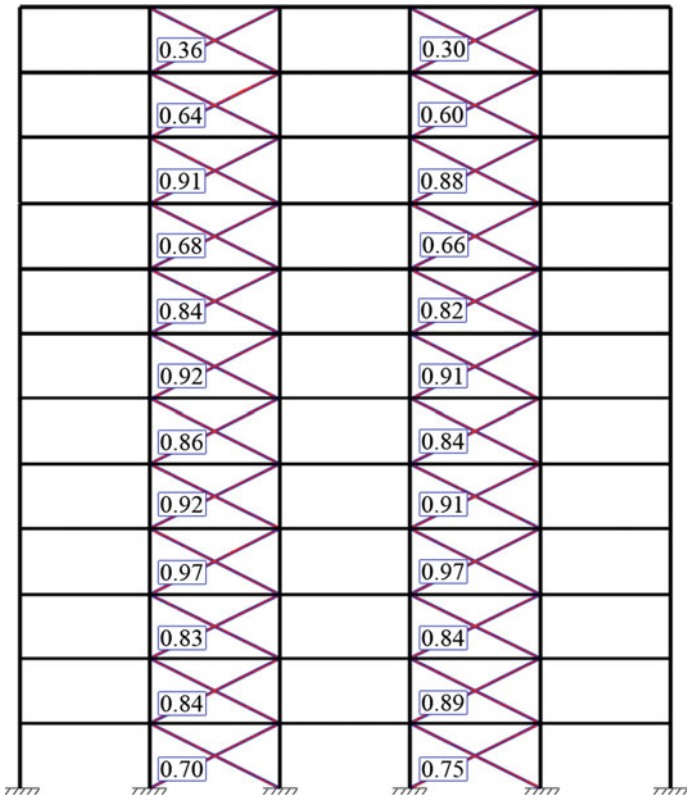

Fig. 7 Δ/Δ_{all} values at IO level for The optimal twelve-story steel dual braced frame found by BA

The reported results in Table 3 exhibit that the optimal solution found by BA is, respectively, 3.27 and 1.56 % lighter than the solutions found by PSO and FA. In addition, FA converges to an optimal solution which is 1.75 % lighter than that of PSO.

Comparison of the convergence histories of Fig. 6 signifies that the BA possesses better convergence behavior with respect to PSO and FA. Furthermore, FA is better than PSO in terms of convergence rate.

In this example, the active constraints which dominate the design are the axial deformation of braces and the lateral drift constraints at IO performance level. For the BA optimal design, the values of Δ/Δ_{all} for all braces and the lateral drift profile at IO level are given in Figs. 7 and 8, respectively.

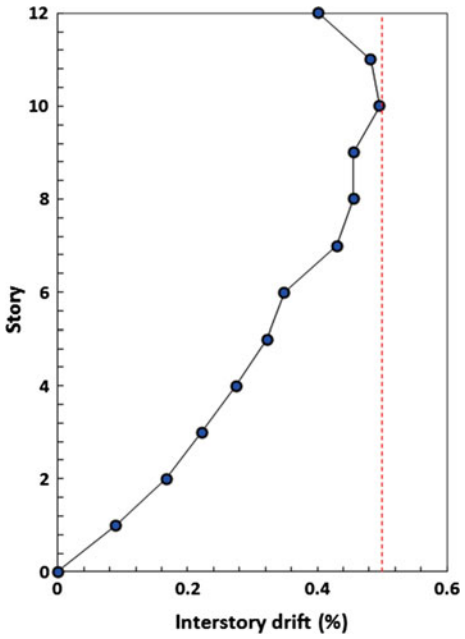

Fig. 8 Drift profile at IO level for the optimal twelve-story steel dual braced frame found by BA

5 Conclusions

This chapter presents a comparative study on the computational performance of popular PSO, FA, and BA meta-heuristic algorithms for tackling the problem of performance-based optimum seismic design of steel dual braced frames. During the optimization process two types of constraints have to be checked. To ensure serviceability of the design, each structural element is checked to satisfy the AISD-LRFD constraints for the non-seismic load combinations. While the second type includes the check of inter-story drifts, plastic rotation of beams and columns and the axial deformation for braces at IO, LS and CP performance levels according to the FEMA-356 provisions for the seismic load combinations. To achieve this, the seismic responses of structures are evaluated by conducting nonlinear pushover analysis.

Two numerical examples including six-story and twelve-story steel dual braced frames are presented. The optimization task is achieved using PSO, FA, and BA meta-heuristics and the results are compared. The numerical results demonstrate that in the first example, BA finds a solution which is 3.62 and 1.74 % lighter than those of the PSO and FA, respectively. The results imply that the solution found by FA is slightly better than the PSO. It is observed that axial deformation constraints for braces at IO performance level dominate the optimal design. In the second example, the optimal weight of BA is 3.27 and 1.56 % lighter than those of the PSO and FA, respectively, and the solution of FA is slightly lighter than that of PSO.

Moreover, the axial deformation of braces and the lateral drift constraints at IO level dominate the optimal designs. In both numerical examples the convergence rate of BA is better than those of the PSO and FA.

It can be finally concluded that for solving the complex and highly nonlinear optimization problem of performance-based seismic design of steel dual braced frames, the BA provides results which are better than those of the PSO and FA meta-heuristics in terms of optimal weight and convergence rate. Therefore, BA can be effectively employed to design cost-efficient steel structures with desirable seismic performance.

References

1. Gholizadeh, S.: Performance-based optimum seismic design of steel structures by a modified firefly algorithm and a new neural network. Adv. Eng. Softw. **81**, 50–65 (2015)
2. Gholizadeh, S., Moghadas, R.K.: Performance-based optimum design of steel frames by an improved quantum particle swarm optimization. Adv. Struct. Eng. **17**, 143–156 (2014)
3. Kaveh, A., Laknejadi, K., Alinejad, B.: Performance-based multi-objective optimization of large steel structures. Acta. Mech. **223**, 355–369 (2012)
4. Fragiadakis, M., Lagaros, N.D.: An overview to structural seismic design optimisation frameworks. Comput. Struct. **89**, 1155–1165 (2011)
5. Liu, M., Burns, S.A., Wen, Y.K.: Multiobjective optimization for performance-based seismic design of steel moment frame structures. Earthq. Eng. Struct. Dyn. **34**, 289–306 (2005)
6. Fragiadakis, M., Lagaros, N.D., Papadrakakis, M.: Performance-based multiobjective optimum design of steel structures considering life-cycle cost. Struct. Multidisc Optim. **32**, 1–11 (2006)
7. Kaveh, A., Farahmand Azar, B., Hadidi, A., Rezazadeh Sorochi, F., Talatahari, S.: Performance-based seismic design of steel frames using ant colony optimization. J. Constr. Steel Res. **66**, 566–574 (2010)
8. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Gonzalez, J.R. et al. (eds.) Nature Inspired Cooperative Strategies for Optimization (NISCO 2010). Studies in Computational Intelligence, vol 284, pp. 65–74. Springer: Berlin (2010)
9. Gholizadeh, S., Shahrezaei, A.M.: Optimal placement of steel plate shear walls for steel frames by bat algorithm. Struct. Des. Tall Spec. Buil. **24**, 1–18 (2015)
10. Hasancebi, O., Carbas, S.: Bat inspired algorithm for discrete size optimization of steel frames. Adv. Eng. Softw. **67**, 173–185 (2014)
11. Hasancebi, O., Teke, T., Pekcan, O.: A bat-inspired algorithm for structural optimization. Comput. Struct. **128**, 77–90 (2013)
12. FEMA-273 (1997) NEHRP guideline for the seismic rehabilitation of buildings. Federal Emergency Management Agency, Washington
13. FEMA-356 (2000) Prestandard and commentary for the seismic rehabilitation of buildings. Federal Emergency Management Agency, Washington
14. McKenna F, Fenves GL (2001) The OpenSees Command Language Manual (1.2. edn). PEER
15. Manual of steel construction: Load and Resistance Factor Design. American Institute of Steel Construction, Chicago, IL (2001)
16. Sullivan, T.J., Calvi, G.M., Priestley, M.J.N., Kowalsky, M.J.: The limitations and performances of different displacement based design methods. J. Earthq. Eng. **7**, 201–241 (2003)
17. Federal Emergency Management Agency: NEHRP guidelines for the seismic rehabilitation of buildings, Rep. FEMA 273 (Guidelines) and 274 (Commentary), Washington (1997)

18. Vanderplaats, G.N.: Numerical Optimization Techniques for Engineering Design: With Application, 2nd edn. McGraw-Hill, New York (1984)
19. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43. IEEE Press, Nagoya (1995)
20. Yang, X.S.: Firefly algorithms for multimodal optimisation. In: Watanabe, O., Zeugmann, T. (eds.) Procedings of the 5th Symposium on Stochastic Algorithms, Foundations and Applications. Lecture Notes in Computer Science, vol. 5792, pp 169–178 (2009)
21. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Mixed variable structural optimization using firefly algorithm. Comput. Struct. **89**, 2325–2336 (2011)
22. Yang, X.S.: Firefly algorithm, stochastic test functions and design optimization. Int. J. Bio-Inspired Comput. **2**, 78–84 (2010)
23. Gholizadeh, S., Barati, H.: A comparative study of three metaheuristics for optimum design of trusses. Int. J. Optim. Civil Eng. **2**, 423–441 (2012)
24. Carbas, S, Hasancebi, O.: Optimum design of steel space frames via bat inspired algorithm. In: 10th World Congress on Structural and Multidisciplinary Optimization, Florida, USA (2013)
25. Gandomi, A.H., Yang, X.S., Alavi, A.H., Talatahari, S.: Bat algorithm for constrained optimization tasks. Neural. Comput. Appl. **22**, 1239–1255 (2013)
26. MATLAB: The Language of Technical Computing. Math Works Inc (2011)

Genetic Algorithms for Optimization of 3D Truss Structures

Vedat Toğan and Ayşe Turhan Daloğlu

Abstract Various optimization techniques have been applied to find the optimum solutions of structural design problems in the last 50 or 60 years. Simple structural optimization problems with continuous design variables have been solved initially using mathematically diverse techniques. New approaches called meta-heuristic techniques have been emerging along with the progress of traditional methods. This chapter first introduces the mathematical formulations of optimization problems and then gives a summary and development process of the preliminary techniques such as genetic algorithm (GA) in obtaining the optimum solutions. The mathematical formulations of the structural optimization problems are associated with the design variables, loads, structural responses, and constraints. Strategies are proposed to improve the performance of the technique to reduce the number of search and the size of the problem. Finally, some examples related to 3D truss structures are presented.

1 Introduction

Optimization of the structures is one of the main research areas in civil and structural engineering. As a branch of applied and computational mathematics, optimization usually tries to find the best-fitted solution of the problem within a domain that contains acceptable values of design variables subject to some design restrictions or constraints. The optimum solution of the problem may be achieved by minimizing or maximizing a real objective function satisfying predefined restrictions at the same time. Such a solution is supposed to be the best one among a large feasible solution space that can satisfy all the constraints of the optimization problem. The function to be minimized or maximized is referred to as objective

V. Toğan · A.T. Daloğlu (✉)
Department of Civil Engineering, Karadeniz Technical University,
61080 Trabzon, Turkey
e-mail: aysed@ktu.edu.tr

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*,
Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_6

function, while the functions to represent restrictions are called as constraints. The design parameters of an optimization problem are called design variables. While the geometric properties of members can be considered as design variables in structural weight optimization, coordinates of nodal points can be treated as design variables for shape or topology design of structures. So the objective function, constraints, and variables can vary widely according to the type of the problem. For the minimum weight design of 3D truss structures, the objective function represents the weight of the truss, variables can be the areas of cross sections of the structural members, and the constraints may be the maximum allowable stresses and displacements of nodal joints.

2 Mathematical Formulations of Optimization Problems

An optimization problem subject to some constraints can be formulated as the following mathematical form:

$$\begin{aligned} &\min_{\mathbf{x}\in S} f(\mathbf{x}) \\ &S = \left\{ \mathbf{x} | h_j(\mathbf{x}) = 0, j = 1, \ldots, p;\ \ g_k(\mathbf{x}) \le 0,\ k = 1, \ldots, m \right\} \end{aligned} \tag{1}$$

where $f(\mathbf{x})$, $h_j(\mathbf{x})$, and $g_k(\mathbf{x})$ are a C^1 function of $\mathbf{x} \in IR^n$. In addition, p and m are the total numbers of $h_j(\mathbf{x})$ and $g_k(\mathbf{x})$, respectively. S represents the feasible set for the optimization problem, $S \subset IR^n$.

However, in the engineering field, an optimization problem can also be generally defined as follows:

$$\begin{aligned} &\text{find} \quad \mathbf{x} = \{ x_1, x_2, \ldots, x_n \} \\ &\text{min.} \quad W(\mathbf{x}) \\ &\text{s.t.} \quad g_k(\mathbf{x}) \le 0, \quad k = 1 \text{ to } m \\ &\qquad x_{il} \le x_i \le x_{iu}, \quad i = 1 \text{ to } n \end{aligned} \tag{2}$$

In Eqs. (1) and (2), $\mathbf{x}$ is an n-dimensional vector representing the design variables of the optimization problem. Depending on the optimization type, the cross-sectional areas of the members, the nodal coordinates of the member connections, and the members itself are treated as the design variables. $f(\mathbf{x})$ and $W(\mathbf{x})$ are called objective functions or cost functions, which usually correspond to a real number to be used to evaluate how good a solution is. Since weight is usually adopted as the objective function of the optimization problem in structural engineering, $W(\mathbf{x})$ corresponds to the weight of the related structure. $h_j(\mathbf{x})$ and $g_k(\mathbf{x})$ are the equality and inequality constraints functions, respectively. In contrast, $h_j(\mathbf{x})$ in the engineering fields, the type of inequality constraint functions, $g_k(\mathbf{x})$, are often encountered. And it generally requires a structural analysis to obtain the structural response such as displacements,

forces, etc. Here, x_{il} and x_{iu} show the lower and upper boundaries of x_i. Strictly speaking, x_i might be continuous, discrete, and real but since it is preferred in practice to select the cross-sectional areas of members from the predefined list, x_i is taken as the discrete in structural engineering applications.

3 Genetic Algorithms

To find the value of $\mathbf{x}$ obeying the $h_j(\mathbf{x})$ and $g_k(\mathbf{x})$ and minimizing $W(\mathbf{x})$ requires an optimization method to be employed. Optimization methods, generally speaking, are classified as the gradient-based and gradient-free techniques. Of course, various categorizations can be encountered for the optimization methods available in the literature [1–3]. In fact, although they are referred with various classifications and names, there is not much difference in the main properties of the methods. For example, one of the most commonly used classifications is deterministic and stochastic. While the former uses derivatives of the objective function and constraints in the search of the optimum solution, the latter works with probabilistic transition rules instead of the gradient information of the objective function and constraints [4–12]. As seen from the example, deterministic and stochastic techniques are easily put into the classification of the gradient-based and the gradient-free optimization methods, respectively.

Genetic algorithm (GA) is probably one of the first optimization methods, which simulates the natural phenomena into a numerical algorithm. It mimics the procedure known as survival of the fittest. GA was firstly proposed by Holland [13]. Since new some valuable improvements in the GA such as adaptive operators, distinct coding schemes for the design variables, immigration, elitism, breeding, hybridization, etc., have been presented by the researcher [19–30].

After the study of Rajeev and Krishnamoorthy [14] which was a well-documented study for the application of the GA in the structural optimization problems, GA has gained more popularity in this area than that proposed for the first time. By studying Rajeev and Krishnamoorthy [14], it is realized at first glance that GA is very primitive compared with the level recently reached. For instance, the design variables were coded with binary scheme in the GA process proposed by Rajeev and Krishnamoorthy [14], while at the moment the design variables are treated as discrete and even that mixed [15–18]. Figure 1 demonstrates the binary scheme for the design variables of the structural optimization problems. Herein, it is assumed that the corresponding optimization problem has three design variables

	1011	0100	0110
$\mathbf{x}$=[	x_1	x_2	x_3]

Fig. 1 Coding in binary system for design variables

which are the cross-section areas of the grouped members. And they are selected from 16 different sections available in practice.

When this scheme is chosen to find the value of the design variables in decimal system, a transformation is needed. Through this transformation, the sequence numbers of the sections related to the design variables are determined from a predefined list sections. The steps of the transformation and mapping can be summarized as the main steps shown in Fig. 2. After mapping with the predefined list, the cross-section area for the related design variable is directly used in the corresponding process, i.e., in calculating the structural responses via a structural analyzer.

Then the structure should be analyzed to determine its responses and then the requirements of the optimization problems defined by the constraints functions are evaluated. The next step is to calculate the value of $W(\mathbf{x})$. As mentioned before, it shows the goodness of the solution that consists of coupling the design variables (see Fig. 1)—the string is called as a solution. However, since the GA is an unconstrained optimization method like other gradient-free optimization techniques, $W(\mathbf{x})$ also includes the value of a function known as the penalty function which reflects the violation level of the constraints in normalized form for the solution. Equation (3) shows the objective function, incorporating the constraints violation as expressed in Rajeev and Krishnamoorthy [14].

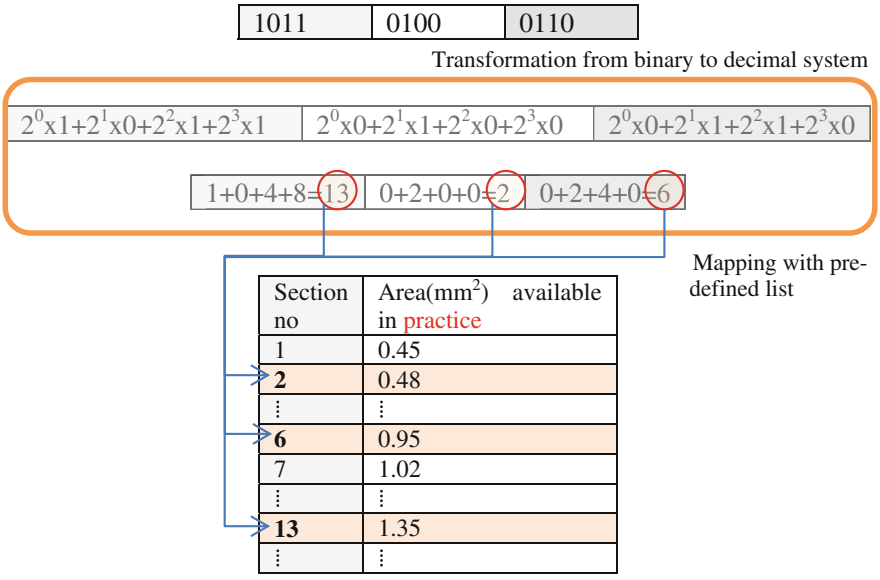

Fig. 2 Decoding and mapping steps for the design variables represented in the binary system

$$W(\mathbf{x}) = W'(\mathbf{x})\,(1 + KC)$$
$$\text{where } C = \sum_{k=1}^{m} c_k \tag{3}$$

where K is a parameter that is taken as 10, C is the violation coefficient computed in the following manner: if $g_k(\mathbf{x} > 0$, then $c_k = g_k(\mathbf{x})$; or if $g_k(\mathbf{x} \leq 0)$, then $c_k = 0$. Finally, $W'(\mathbf{x})$ represents the weight of the structure. Note that $g_k(.)$ is in the normalized form expressed as $\sigma/\sigma_a - 1 \leq 0$ for stress and $d/d_a - 1 \leq 0$ for displacement. σ_a and d_a in these expressions show the allowable values for stress and displacement for the problem considered.

3.1 Genetic Operators

The search procedures of GA were based on the mechanics of natural genetics and natural selection. With the help of the genetic operators adapted from nature, the concept of the survival of the fittest is simulated to form a robust search mechanism. Following the steps summarized above, the GA search procedure proposed by Rajeev and Krishnamoorthy [14] applies two genetic operators successively.

The first one is the reproduction operator which reflects the concept of the survival of the fittest in nature. It proceeds according to the individual fitness calculated as follows:

$$F_i = (W_{\max} + W_{\min}) - W_i(x) \tag{4}$$

where F_i is the fitness of the ith individual in the population, $W_{\max}$ and $W_{\min}$ are the maximum and minimum values of $W(\mathbf{x})$ computed using Eq. (3). Thus, the individuals with higher fitness values have a higher probability to survive, whereas the less fit ones get fewer chances of survival. And the worst fit individuals will be removed from the population.

Then, to exchange the solution segments between the individuals in the population, crossover operator is implemented. Double point crossover is applied to the pairs selected randomly as an example. Figure 3 demonstrates the application of the two genetic operators explained in Rajeev and Krishnamoorthy [14].

The genetic algorithm continues the process by following the above steps outlined in detail so as to modify the new population. To terminate the GA process, a criterion based on the similarity of the individuals in the population was imposed by Rajeev and Krishnamoorthy [14]. In addition, although mutation operator was not implemented in [14], the GA search process employed it to preserve the diversity among the population. Figure 4 illustrates the basic concept behind the mutation operator. It proceeds in three steps. First, an individual within the population is randomly selected. Then, a binary position to be changed is determined randomly.

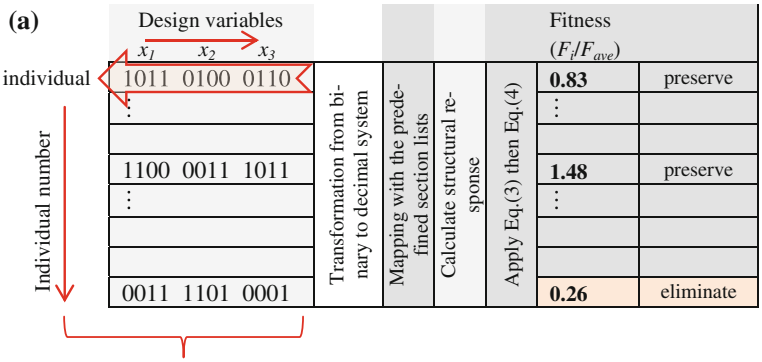

Fig. 3 Two genetic operators given in Rajeev and Krishnamoorthy [14]. **a** Reproduction operator. **b** Crossover operator

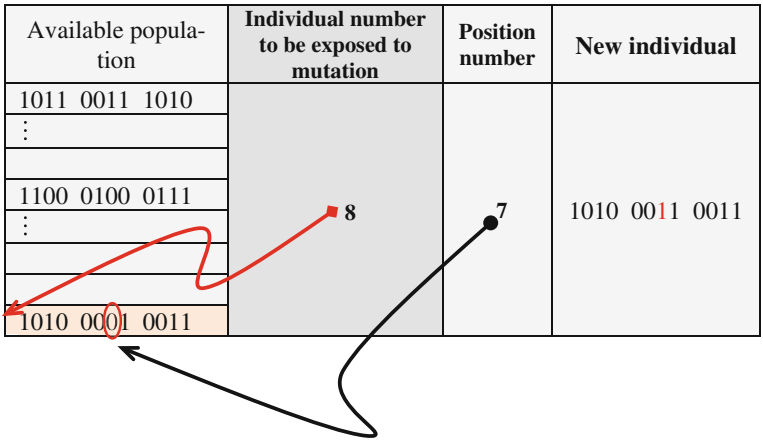

Fig. 4 Mutation operator

After that, the corresponding position value is switched to 1 if it is equal to 0 or to 0 if it is equal to 1.

All search procedures based on GA were called simple genetic algorithms in Rajeev and Krishnamoorthy [14] and since then, numerous modifications have been proposed by researchers in order to improve the search and/or computational performances of the GA in this area. For example, the coding scheme [19–21], the penalization [22–27], and new operators [28–30] are some of them. Other improvements based on adaptive concept have attracted more attention. The following section will review this concept.

4 Strategies Based on the Adaptive Concept

Some improvement or renewal in the GA operators or the GA algorithms have been made by researchers to increase the probability of finding the global solution and to enhance the performance of GA. Other improvements of GA are to relieve the user from the burden of determining sensitive parameter(s) existed in GA. The vast majorities of these efforts have been focused on the adaptive approaches in GA for both the penalty function, and the mutation and crossover. The key idea behind the adaptive approaches is to adjust itself automatically during the optimization procedure using genetic algorithms.

4.1 Adaptive Penalty Scheme

Although it is not a genetic operation, the penalty function is important in GA to demonstrate the extent of the violation of the constraints quantitatively. Penalty techniques can be classified as multiplicative or additive. A positive penalty factor is introduced in the multiplicative case to amplify the value of the fitness function of an infeasible individual in a minimization problem. This type of penalty has received less attention in the evolutionary computation community, compared with the additive type. A penalty functional is added to the objective function in the additive case to define the fitness value of an infeasible element [24].

An adaptive penalty scheme to be able to adjust itself automatically during the GA process is proposed by Toğan and Daloğlu [31] as follows:

$$\begin{aligned} f_{\text{penalty}} &= (C_{\max} + C(r))/(C_{\max} + C_{\text{ave}}) \qquad C(r) \geq C_{\text{ave}} \\ f_{\text{penalty}} &= (C_{\text{ave}} + C(r))/(C_{\text{ave}} - C_{\min}) \qquad C(r) < C_{\text{ave}} \\ f_{\text{penalty}} &= 0 \qquad\qquad\qquad\qquad\qquad C(r) = 0 \qquad r = 1, \ldots, \text{nps} \end{aligned} \tag{5}$$

where f_{penalty} is the penalty function, $C(r)$ is the violation value of normalized constraints of the rth individual in the generation, and nps represents the population

size. In addition, $C_{\max}$, $C_{\min}$, and C_{ave} are, respectively, the maximum, minimum and average violation values of generation. According to this formulation, Eq. (3) becomes

$$W(\mathbf{x}) = W'(\mathbf{x})\left(1 + f_{\text{penalty}}\right). \tag{6}$$

The penalty function is, therefore, kept free from any predefined or user-defined constants. Also the magnitudes of the violations are not characterized by a static rate for both near feasible and infeasible solutions during the design process. With the expressions in Eq. (5), the infeasible solutions will not be penalized with the same rate of penalty. The magnitude of the penalty tends to get heavier instead, as the level of the violation of the infeasible solution tends to get bigger. Moreover, the magnitude of penalty increases as the violation value gets closer to $C_{\max.}$ On the other hand, it decreases as the violation value gets closer to $C_{\min.}$ Thus, some infeasible individuals that are close to the feasible region in the search space will not disappear through the penalty scheme and they will find a chance to survive. This may sustain the capacity of finding the global solution for design problem using GA.

4.2 Adaptive Crossover and Mutation Schemes

Genetic operators are applied to mimic the natural evolution. Among these operators, crossover provides the genetic information exchange between the couples randomly, and mutation enables the development of new genetic material, and both play an effective role to reach the optimum or to get close to the optimum solution. It is arbitrary and up to the user to incorporate the rates of these operators in optimization process. The choices of mutation, p_m, and crossover, p_c, rates as well as generating positions to be shifted by mutation and mapping the individuals for crossover are also arbitrary.

Many refinements using adaptive controls provide significant improvements in performance for some situations [32]. Keeping all of these in mind, it can be concluded that the randomness on mapping for crossover, and specifying the gene position(s) for mutation may be removed. Mutation and crossover should be adapted for both the individual and the generation because it is possible to lose the best-fitted individual with this random process in mutation. Therefore, in contrast to traditional crossover and mutation operator based on randomization mechanisms, i.e., generating the pairs, and determining position of bit shifted by mutation of the solution, the mutation and the crossover operators can be adaptive and adjust themselves from generation to generation since the population is renewed from iteration to iteration. Adaptive means adjusting itself automatically depending on the fitness value of the individual and the other individuals in the generation.

The adaptive mutation and crossover operators suggested by Srivinas and Patnaik [33] are modified as follows and applied by Toğan and Daloğlu [31, 34–36], for the optimization of 2d and 3d truss structures:

$$p_m = \begin{cases} 0.5(W_{\max} - W(r))/(W_{\max} - W_{\text{ave}}) & W(r) \geq W_{\text{ave}} \\ (W_{\text{ave}} - W(r))/(W_{\text{ave}} - W_{\min}) & W(r) < W_{\text{ave}} \end{cases} \tag{7a}$$

$$p_c = \begin{cases} (W_{\max} - W^*)/(W_{\max} - W_{\text{ave}}) & W^* \geq W_{\text{ave}} \\ 1 & W^* < W_{\text{ave}} \end{cases} \tag{7b}$$

Here, $W(r)$ is the fitness of the rth individual, W_{ave} is the average fitness value of the population, $W_{\max}$ and $W_{\min}$ are the maximum and minimum fitness values of an individual in the population, respectively, and W^* is the larger of the fitness values of the solutions to be crossed. p_m and p_c are mutation and crossover rates, respectively.

After the mutation rate, p_m, is determined using Eq. (7a), the numbers of design variables, m_{des}, disrupted by mutation are calculated by multiplying p_m with the string length of the solution. Then design variables in the individual are arranged according to the level of violation of normalized constraints, and they are renewed with m_{des} starting with the most violated one. Thus design variables in the individuals are classified and the good individuals are kept unchanged. Also the diversity of population is maintained since the design variables that violate the constraints are renewed.

Unlike Srivinas and Patnaik [33], and Bekiroğlu [37], W^* represents the lower value of the two fitness values of the solutions for crossover. The reason for that is because if the lower value of fitness is bigger than W_{ave}, the crossover will take place between the pairs having good fitness value, whereas when W^* represents the higher of the fitness values, there is a possibility for crossover to take place between the pairs having bad fitness values. Since the adaptive crossover is incorporated, information exchange between pairs can be done with various crossover points changing from 1 to string length of the individual (flexible point crossover). The numbers of design variables, c_{des}, exchanged by crossover between pairs are specified by multiplying p_c with string length for the solution. So if p_c is equal to 1, the individual of pairs will not be subjected to crossover operator.

5 Innovative Approaches in Genetic Algorithms

The enhancements developed by the researchers to increase the performance of GA are not limited to adaptive schemes. Besides, some refinements are proposed for making the search procedures of GA computationally effective. Member grouping and initial population strategies are some of them as described in the next subsections in detail.

5.1 Initial Population Strategy

To start the evolution process in GA, the initial population within the solution space is generated. All individuals constituting the initial population are selected from the solution space of the problem notwithstanding any condition. In other words, this step is random. However, even though the process of creating initial population seems ordinary, it critically affects the convergence, the performance and the ability of the GA. This case becomes more crucial for very complicated and large solution space which is mostly encountered in the practical application of GA in the area of structural engineering.

The idea starting the search of the solution space without a randomly generated set is the key rationale of creating the initial population automatically. So, adopting the list number of the maximum area of cross sections as the starting point for the design variables leads to more efficiency than randomly generated. And it is stored as the initial point for each group of tension members to create the initial population. For the groups of the compression members, two or three surplus of the list number of the member that has he maximum area of cross section in the group is taken from the list of sections. The value of cross-sectional area and radius of gyration of that section must be bigger than the values found previously. And the corresponding section list number gives the initial points for each group of compression members and is stored to create the initial population (see Toğan and Daloğlu [31, 34] for more details).

5.2 Member Grouping Strategies

In the structural optimization terminologies, the design variables refer to the variables affecting the value of the objective function, and they generally represent the areas of the cross section of the structural members. Since the GA completely independent from the characteristic of the problem, the design variables of the optimization problems should be coded in some encoding schemes such as binary, decimal, real, and so on. GA evolves those solutions by creating in terms of design variables and randomly selecting into the potential solutions space of the problem. A set of possible feasible or unfeasible solutions construct the population or generation. Each solution in the populations is known as the individual.

For a given problem, all of the cross-sectional areas of the structural members can be taken as design variables. In this case, however, the computation time gets very high and the results obtained from optimization process will probably be the local optima due to the expanded design space. Therefore in the GA applications, member grouping is generally applied for the members of the structural system in order to reduce the size of the problem. On the other hand, the member grouping adopted a priori might not lead to an accurate grouping and if the number of members of the structural systems becomes very large; i.e., for 3D roof trusses,

transmission towers, this leads to very large string lengths, which delays convergence and precludes useful exchange of information [38].

Two new member grouping strategies are proposed by Toğan and Daloğlu [31, 34] to reduce the size of the search space of the design problem as much as possible, to increase the probability of catching the global optimum solution and enhance the performance of the GA. The key idea behind the member grouping strategies is to make a convenient member grouping that will end up with as few numbers of cross sections as possible in the final set, and reduce the size of the design space of the problem as much as possible. The efforts are also made to relieve the user from the burden of determining the member groups.

5.2.1 First Member Grouping Strategy

The first strategy (strategy 1) is based on the one proposed by previous researchers [31, 38, 39]. To implement this strategy, the same cross-section areas are assigned for all the structural members first as stated in [31, 38, 39]. Then the analysis of the structure is performed using these initial areas for each load cases. Following the static analysis, the entire range of internal forces is divided into several ranges both for tension and compression members. And members are grouped according to the internal forces in the members.

An additional group is added in Toğan and Daloğlu [34] to the system for zero force members or members with very low internal forces. Hence, all the members of the truss structure are grouped conveniently and accurately. Moreover, a complex solution space may be avoided under some conditions.

5.2.2 Second Member Grouping Strategy

For strategy 2, while the magnitude of the axial force is considered as the factor for grouping the tension members [31, 38, 39], slenderness ratio is considered as the main factor for the compression members to set the groups. Therefore, due to the importance of slenderness ratio, it may be more convenient to group the compression members according to their slenderness ratio in terms of radius of gyration of the cross section and the effective length of the member instead of grouping them depending on the magnitude of the axial force. This is the key idea behind strategy 2. Hence, as the tension members of the truss structure are grouped depending on the axial forces, the compression members are grouped according to their slenderness ratio. An extra member group for the zero force members or members with very low internal forces is also arranged (Toğan and Daloğlu [31, 34]).

6 Examples

In this section, in the light of the information given in the previous sections, two types of design examples are presented. In the first one, a 160-bar space truss is considered as an example to demonstrate the effectiveness and robustness of proposed adaptive approaches for GA and member grouping strategy over simple GA. Later, an investigation is performed to demonstrate the efficiency, accuracy, and reliability of the proposed initial population strategies by solving numerical examples taken from previous studies in the literature for comparison.

The population size is taken as 40 for all the examples and real-value coding is employed in the genetic algorithm.

At the beginning of the genetic process 40 % of the initial population is created by using the proposed initial population strategy automatically. Therefore the diversity of the population is preserved and the algorithm may be less likely to get stuck at local minima and may also avoid some early convergence. It is possible to create all the individuals of the initial population automatically. However, in this case, the initial population consists of the same individual only and the search performed in the solution space start in a certain region. On the other hand, as the adaptive schemes applied for both penalty functions and mutation and crossover operators are able to adjust itself automatically during the genetic process [31], they completely disrupt the initial population. So, creating all the individuals in the initial population automatically is not meaningful.

6.1 Example 1: 160-Bar Truss Tower

The 160-bar truss tower shown in Fig. 5 was optimized by Rajaev and Krishnamoorthy [14] and Galante [40] in advance. 32 cross-section types were used to optimize the tower and taken from the AISC Manual [41]. The members are classified into 16 groups. Details of the member groups were presented in Galante [40]. Rajaev and Krishnamoorthy [14] used in GA with one criterion (minimum weight) and without taking the buckling effect into account. As Galante [40] stated, it can be observed that the buckling effect plays an important role in truss optimization. So if it is not taken into account the truss obtained will not be suitable as load carrying structural system. Galante [40] optimized the transmission tower with the aim of the minimum weight and minimum number of cross-section types of bars taken from the market and also taking buckling and the slenderness limits recommended by AISC [41] into account.

The tower is optimized for the objective indicated by Galante [40] with the proposed algorithm. All parameters needed to start the optimization process are taken from the reference studies. The structure is optimized with the implemented improvements in GA and two member grouping strategies for the aim of that the final design forms three groups at one for tension and two for compression

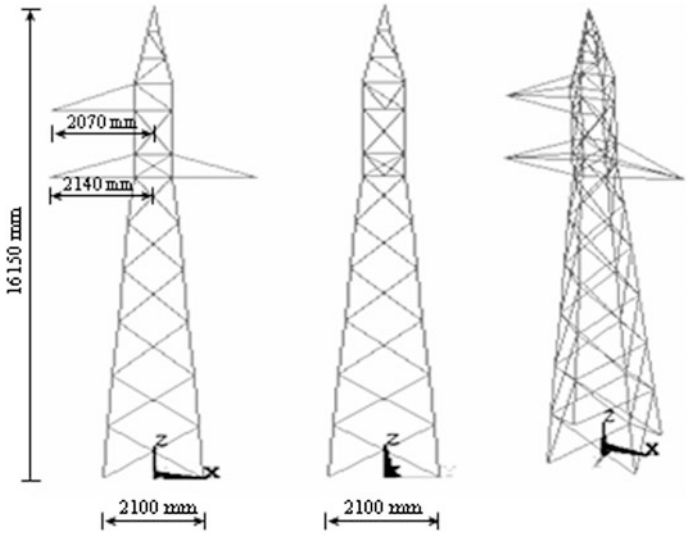

Fig. 5 160-bar trussed steel transmission tower

members. An extra member group for zero force members is not formed. This is because of the fact that Galante [40] reported that 51 bars have a higher compression stress than the limit recommended by AISC [41] to prevent buckling and 68 have higher slenderness ratio than the AISC advice. It is also observed from the optimization process of the tower that the arrangement of an extra member group for zero force members does not make any difference in the weight of the tower drastically and the two member grouping strategies give the same result. Therefore, the tower is optimized with three, four, and five member groups.

The final design obtained by the proposed strategies and the ones reported by Galante [40] are presented in Table 1. Galante [40] used both GA and the simple rebirth process in GA for the optimization of this example and mentioned that the GA with the rebirth process achieves a better truss. However, the optimum designs obtained with the proposed improvements in the algorithm ended up with a lighter truss than the design by Galante [40]. A question may arise: as the values of design variables presented in Table 1 for the three groups are the same, why are the optimum weight of truss different? The answer is hidden in Fig. 6a. It is shown in Fig. 6a that some members that belong to the first group skipped to the second group and a better solution from the previous one is reached. However, the most interesting result is obtained when the four member groups are adopted at two for tension and two for compression member for this example. Although this optimum design has more groups than the result obtained by Galante [40] and previously performed optimization cases, in this study, it seems still reasonable to achieve that the number of sections in the final set must be as few as possible to make fabrication and workmanship easier [31, 39, 40]. The reduction in the total weight of the

Table 1 Comparison of results for 160-bar transmission tower

Design variables (mm^2)	Galante [40] at 3 groups by the GA	Galante [40] at 3 groups by the GA with rebirth	This study			
			At 3 groups	At 3 groups	At 4 groups	At 5 groups
A_1		2812	767.74	767.74	767.74	767.74
A_2		3064	581.93	581.93	339.99	339.99
A_3		7096	1251.61	1251.61	581.93	581.93
A_4					1251.61	1251.61
A_5						150.96
Weight (kN)	15.33	14.61	14.269	12.651	10.544	10.449

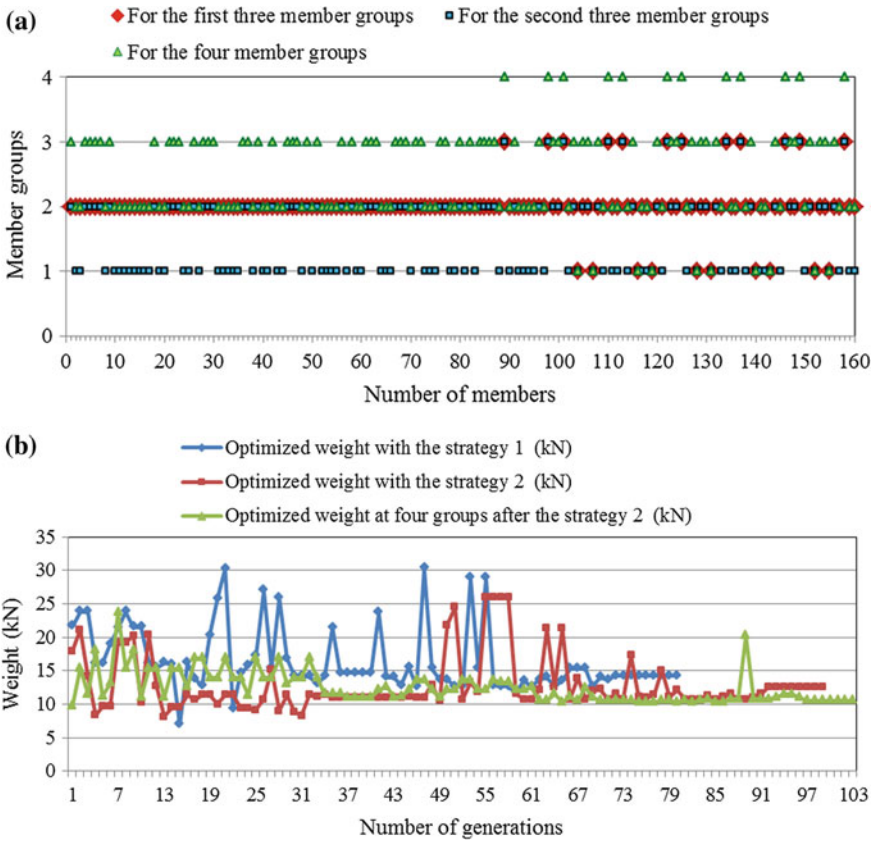

Fig. 6 **a** Member groups for 160-bar tower, **b** Variation of weight for the three cases with number of generations

tower is about 28 % compared to the designs summarized in Table 1. As mentioned above, when the number of groups is taken as five, the optimized weight is slightly changed. In other words, arranging an extra group for the zero force members did not affect the weight of the tower drastically. Hence, the design at four groups is more meaningful than the design at five groups for the design objective. Figure 6b shows the genetic histories of the value of the objective function taken as the weight of tower for three optimization cases.

6.2 *Example 2: 640-Bar Space Deck Truss*

A trussed space deck shown in Fig. 7 has 179 joints and 640 members. The truss was first studied by Jenkins [32] in order to assess the decimal GA on a more substantial structural problem. Jenkins [32] found the optimum height of truss in addition to optimum volume. The truss members were subjected to compressive stress limitation given in BS 5950 and tensile stress limitation, 275 N/mm^2. This trussed space deck was subjected to the one loading condition, which a single load of 300 kN was applied at the center of the upper plane (joint 90). A maximum displacement limitation of ±40 mm was imposed on every node in vertical direction. 21 discrete values of data for each design variable were taken from rectangular hollow steel sections with cross-sectional areas varying from 142 mm^2 in intervals to 4350 mm^2. Jenkins [32] collected the members of the structure in five distinct groups as follows: (1) upper deck longitudinal members, (2) upper deck transverse members, (3) lower deck longitudinal members, (4) lower deck transverse members, and (5) diagonal members connecting the upper and lower planes.

For this example, 23 pipe sections given in AISC [41] are adopted for each design variables. The allowable tensile stress is taken as 275 N/mm^2 and the modulus of elasticity is 210 kN/mm^2. The allowable compressive stress is calculated according to AISC [41] for the compression members and the maximum deflection imposed is 40 mm. The truss is subjected to one loading condition as specified in [32]. The optimization of the space truss is first carried out by using five groups imposed by Jenkins [32]. Then the optimal volume of the truss is obtained with four groups of members that were assumed as one for tension, two for compression members, and one for zero force members after preliminary analysis. Table 2 shows the optimum volume for each trial and the maximum deflection of the truss. The algorithm proposed in this study achieved a design with the best solution vector after approximately 29,000 searches for each trial. When four groups for the members are adopted after the preliminary analysis, the volume of the truss gets smaller than the result obtained by using five groups imposed by Jenkins [32]. Moreover, nearly 50 % reduction is obtained with both the member grouping and the new initial population strategies adopted in this study. This design obtained with the proposed algorithm confirms the intension of "both the weight of the structure and the number of cross section should be minimized to obtain an economical structure" as indicated in [31, 33–36, 38–40].

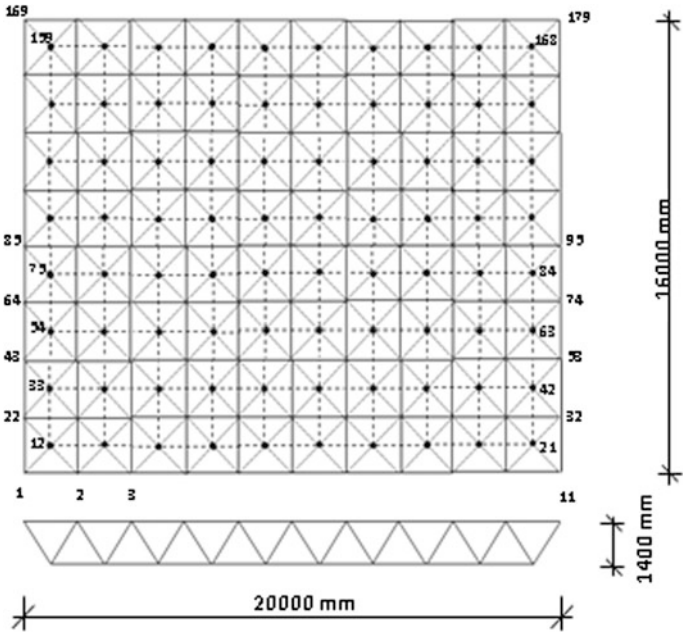

Fig. 7 640 trussed space deck truss

Table 2 Results for the 640-bar space deck truss

Variables	Optimal cross-section areas (mm^2)	
	Case 1	Case 2
A_1	1096.77	568.39
A_2	954.84	690.32
A_3	954.84	1096.77
A_4	690.32	161.29
A_5	1096.77	
Volume (cm^3)	1298026.52	659970.80
Max. def. (mm)	29.03	39.85
Note The coded values design variables for the automatically created individuals		
For case 1		
Coded values	Volume (cm^3)	Violation (stresses + displacements)
12 10 9 8 10	912520	10.503
For case 2		
Coded values	Volume (cm^3)	Violation (stresses + displacements)
9 9 12 1	602088.6	47.84

6.3 Example 3: 1512-Bar Shed Truss

In order to demonstrate the performance, efficiency, and practical capability of the algorithm with the proposed strategies, the design of 1512-bar shed truss is studied as a final design problem. A shed structure illustrated in Fig. 8 covers a square field with size of 40,000 × 40,000 mm^2 and has a height of 5359 mm. The structure is designed in a way that its sub-parts are reproduced along direction Z per 5000 mm and are connected by lateral elements on the lower arch level in that direction. In a sub-part each node on lower arch connects to four nodes on upper arches with diagonals. So, it consists of 409 nodes and 1512 truss elements. The structure is supported at two edges in the Z direction. The material density and modulus of elasticity are 7.85 × 10–8 kN/mm^3 and 210 kN/mm^2, respectively. It is subjected to a load scheme that is applied to all nodes of upper arches. 35,000 N is applied the nodes along Z direction and it increases 10,000 N per each node level in Y direction so that at the two nodes near symmetry axis the value is 55000 N (see Fig. 8). The allowable value of 250 N/mm^2 is employed for tensile stresses and the formulation of buckling obeying AISC [41] considered for compressive stresses. The algorithm is provided with 26 discrete values of data for each design variable. The structural properties are taken from the pipe sections as given in [41]. The maximum displacement limitation imposed is 50 mm.

As seen in Fig. 8, since the shed truss is a large structural system with more than 1000 members, it might be very difficult to optimize the area of individual members if the member groups become very large. Besides, very large string length causes the solution space of the problem to increase and becomes very complicated. Therefore, member grouping adopted to reduce the size of the problem and initial points to search the solution space are crucial to get an optimum design which is closer the global optimum.

The truss is designed by adopting the four groups at one for tension, two for compression, and one for zero members after the preliminary analysis. Table 3 gives the best solution vectors and the corresponding weight. An optimal structural

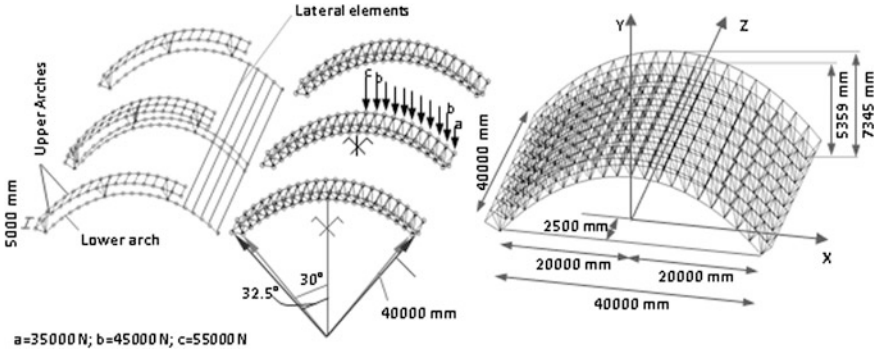

Fig. 8 1512-bar shed truss

Table 3 Result for the 1512-bar shed truss

Variables	Optimal cross-section areas (mm^2)	
A_1	690.32	
A_2	2045.16	
A_3	5419.34	
A_4	515.48	
Weight (kN)	692.43	
Max. def. (mm)	49.31	
Note The coded values design variables for the automatically created individuals		
Coded values	Weight (kN)	Violation (stress + displacement)
10 17 23 8	871.78	8.893

weight of 692.43 kN was obtained considering the constraints, which gives a feasibility to construct it in practice. The algorithm obtained the optimum solution after approximately 43,600 searches.

7 Conclusion

An optimum design approach is proposed based on the GA for truss structures with the help of self-adaptive strategies for member grouping, penalty function, mutation and crossover operators, and the initial population, which is then employed for the optimization of large 3D truss structures. An investigation has been performed to demonstrate the performance and workability of the enhanced GA (eGA). It is shown from the design examples that the eGA works well for the large structural systems. It is also worth pointing out that self-adaptive strategies in eGA help user to start the GA process automatically. It can be expected that proposed eGA may also be a useful search technique and a tool for solving discrete sizing variables of the large 3D truss structures.

References

1. Horst, R., Pardolos, P.M.: Handbook of global optimization. Kluwer Academic Publishers, Dordrecht (1995)
2. Nocedal, J., Wright, J.S.: Numerical optimization. Springer, New York (1999)
3. Chong, E.K.P., Zak, S.H.: Introduction to Optimization. Wiley, New York (2002)
4. Paton, R.: Computing with Biological Metaphors. Chapman & Hall, London (1994)
5. Adami, C.: An Introduction to Artificial Life. Springer, New York (1998)
6. Matheck, C.: Design in Nature: Learning from Trees. Springer, Berlin (1998)
7. Mitchell, M.: An Introduction to Genetic Algorithms. The MIT Press, Cambridge (1998)
8. Flake, G.W.: The Computational Beauty of Nature. MIT Press, Cambridge (2000)

9. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
10. Glover, F., Kochenberger, G.A.: Handbook of Metaheuristics. Kluwer Academic Publishers, Dordrecht (2003)
11. Dreo, J., Petrowski, A., Siarry, P., Taillard, E.: Meta-Heuristics for Hard Optimization. Springer, Berlin (2006)
12. Sean, L.: Essentials of Metaheuristics (2015). http://cs.gmu.edu/∼sean/book/metaheuristics/Essentials.pdf
13. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Co., Reading (1989)
14. Rajeev, S., Krishnamoorthy, C.S.: Discrete Optimization of Structures Using Genetic Algorithms. J. Struct. Eng. **118**(5), 1233–1250 (1992)
15. Tang, X.,·Bassir, D.H., Zhang, W.: Shape, Sizing Optimization and Material Selection Based on Mixed Variables and Genetic Algorithm. Optim Eng **12**, 111–128 (2011)
16. Ahmadi, M., Arabi, M., Hoag, D.L., Engel, B.A.: A mixed discrete-continuous variable multiobjective genetic algorithm for targeted implementation of nonpoint source pollution control practices. Water Resour. Res. **49**, 8344–8356 (2013)
17. Yuan, Q.K., Li, S.J., Jiang, L.L., Tang, W.Y.: A mixed-coding genetic algorithm and its application on gear reducer optimization. Fuzzy Info. Eng. **2**(AISC 62), 753–759 (2009)
18. Rao, S.S., Xiong, T.: A hybrid genetic algorithm for mixed-discrete design optimization. J. Mech. Des. **127**(6), 1100–1112 (2004)
19. Kumar, A.: Encoding scheme in genetic algorithm. Int. J. Adv. Res. IT Eng. **2**(3), 1–7 (2013)
20. Kumar R, Jyotishree (2012) Novel encoding scheme in genetic algorithms for better fitness. Int. J. Eng. Adv. Tech. **1**(6), 214–218
21. Zhu, J., Zhou, D., Li, F., Fu, T.: Improved real coded genetic algorithm and its simulation. J. Softw. **9**(2), 389–397 (2014)
22. Nanakorn, P., Meesomklin, K.: An adaptive function in genetic algorithms for structural design optimization. Comp. Struct. **79**(29–30), 2527–2539 (2001)
23. Kramer, O., Schwefel, H.P.: On three new approaches to handle constraints within evolution strategies. Nat. Comp. **5**, 363–385 (2006)
24. Lemonge, A.C.C., Barbosa, H.J.C.: An adaptive penalty scheme for genetic algorithms in structural optimization. Int. J. Numer. Meth. Eng. **59**, 703–736 (2004)
25. Coello, C.A.C.: Use of a self-adaptive penalty approach for engineering optimization problems. Comp. Ind. **41**, 113–127 (2000)
26. Lin, C.H.: A rough penalty genetic algorithm for constrained optimization. Inform. Sci. **241**, 119–137 (2013)
27. Lemonge, A.C.C., Barbosa, H.J.C., Bernardino, H.S.: A family of adaptive penalty schemes for steady-state genetic algorithms. Proceeding in WCCI 2012, June, pp. 10–15. Brisbane, Australia (2012)
28. Kaya, M.: The effects of two new crossover operators on genetic algorithm performance. Appl. Soft Comput. **11**, 881–890 (2011)
29. Thanh, P.D., Binh H.T.T., Lam, B.T.: New mechanism of combination crossover operators in genetic algorithm for solving the traveling salesman problem. Knowl. Syst. Eng. (AISC 326), 753–759 (2015)
30. Deep, K., Thakur, M.: A new mutation operator for real coded genetic algorithms. Appl. Math. Comput. **193**(1), 211–230 (2007)
31. Toğan, V., Daloğlu, A.T.: Optimization of 3d trusses with adaptive approach in genetic algorithms. Eng. Struct. **28**, 1019–1027 (2006)
32. Jenkins, W.M.: A decimal-coded evolutionary algorithm for constrained optimization. Comput. Struct. **80**(5–6), 471–480 (2002)
33. Srivinas, M., Patnaik, L.M.: Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans. Syst. Man Cybern. **24**(4), 656–667 (1994)
34. Toğan, V., Daloğlu, A.: An improved genetic algorithm with initial population and selfadaptive member grouping. Comput. Struct. **86**, 1204–1218 (2008)

35. Toğan, V., Daloğlu, A.: Adaptive approaches in genetic algorithms to catch the global optimum. Proceeding in ACE 2006, October, pp. 11–13. İstanbul, Turkey (2006)
36. Toğan, V., Daloğlu, A.: optimization of truss systems with metaheuristic algorithms and automatically member grouping. Proceeding in 4th National Steel Structures Symposium, October, pp. 24–26. İstanbul, Turkey (2011)
37. Bekiroğlu, S.: Optimum design of steel frame with genetic algorithm (in Turkish). M.Sc. thesis, Karadeniz Technical University (2003)
38. Krishnamoorthy, C.S., Venkatesh, P.P., Sudarshan, R.: Object-oriented framework for genetic algorithms with application to space truss optimization. J. Comput. Civil Eng. **16**, 66–75 (2002)
39. Sudarshan, R.: Genetic algorithms and application to the optimization of space trusses. A Project Report, Madras (India), Indian Institute of Technology (2000)
40. Galante, M.: Genetic algorithms as an approach to optimize real-world trusses. Int. J. Numer. Meth. Eng. **39**, 361–382 (1996)
41. American Institute of Steel Construction (AISC).: Manual of steel construction-allowable stress design, 9th edn. Chicago (1989)

Hybrid Meta-heuristic Application in the Asphalt Pavement Management System

Fereidoon Moghadas Nejad, Ashkan Allahyari Nik and H. Zakeri

Abstract This chapter presents a hybrid meta-heuristic method which combines particle swarm optimization (PSO) and genetic algorithm (GA) search procedures to predict the pavement condition index (PCI) based on Surveyed Inspection Units (SIUs). Both PSO and GA are used and a comparison is made among three approaches for evaluating the optimal arrangement of SIUs. A hybrid method was developed to build and optimize the models. The performances of these hybrid models were compared based on Sampling Error (SE), Total Network Inspection Error (TNIE), inspection time based on CPU time (seconds), total number of SIUs, and others. Based on the results of the computational experiments, one of the proposed heuristic procedures is used for solving problems in the arrangement of surveyed asphalt pavement inspection units. The study reveals that the hybrid model outperforms both the PSO and the GA based models.

Keywords Pavement management · Pavement condition index · Surveyed inspection units · Particle swarm optimization

F. Moghadas Nejad (✉)
Department of Civil and Environment Engineering, Amirkabir University of Technology, No. 424, Hafez Ave., Tehran, Iran

A. Allahyari Nik
Department of Civil Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

H. Zakeri (✉)
Amirkabir Artificial Intelligence and Image Processing Lab (Attain), Department of Civil and Environment Engineering, Amirkabir University of Technology, No. 424, Hafez Ave., Tehran, Iran
e-mail: h-zakeri@aut.ac.ir

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*, Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_7

1 Introduction

The pavement is an important transportation infrastructure. Therefore, pavement deterioration must be managed and optimal maintenance and rehabilitation (M&R) actions must be selected the over a planning horizon [1–3].

These decisions are supported by an efficient system called the pavement management system (PMS). PMS incorporates data collection, monitoring of the current pavement characteristics, predictions of the future conditions, and prioritization of the M&R actions [4, 5].

Pavement conditions are a key component of PMS as transportation agencies require detailed and timely information about a pavement network following pavement inspection [6, 7]. The pavement inspection process is a basic and important process in PMS for evaluating the true condition and selecting the right M&R actions [8]. An inspection method consists of visual detection and distress assessment by inspectors in a field survey [9]. In the visual inspection the branches are divided into sections and the sections into smaller units as inspection units which are also called sample units, respectively [10]. The inspection process is generally costly and time consuming, and it depends on the experiences of the inspectors. The inspectors decide whether or not all the inspection units need to be surveyed based on the inspection manual, and the budget policies of related agencies [11].

Therefore, two limitations characterize the inspection process. First, to survey all the inspection units requires great effort and expertise, which is very costly and time consuming. Second, budgets for M&R actions are constrained. To overcome these limitations it is required that the inspection process is conducted by selecting a specific number of inspection units as surveyed inspection units (SIUs) that can lead to an acceptable estimation of pavement conditions. Estimation of pavement conditions is based on composite indicators that consider multiple factors, such as traffic, climate changes, and changes in the characteristics of the composing materials. These indicators are defined by related agency for selecting M&R strategies [12], each of them considering the threshold values for variations in the pavement sections [13].

Several example of these indicators include the present serviceability index (PSI), the international roughness index (IRI), the pavement condition rating (PCR), the pavement structural condition (PSC), the present serviceability rating (PSR), the pavement quality index (PQI), distress manifestation index (DMI), and the pavement condition index (PCI) [14–16].

PCI is a more practical indicator than the others because it considers the characteristics of distress such as the type, severity, and extent. The application of the PCI is covered in several works [2, 12, 17–22].

There are two challenges in evaluating the PCI of sections. First, the selection of the low SIUs could not define the accurate condition. Second, the survey of the inordinate number of inspection units requires sufficient expenditure and time, which can lead to delays in execution of M&R actions. It is therefore necessary to

optimize the arrangement of SIUs by proposing a novel method that can simulate the inspection process and evaluate the acceptable accurate PCI in the sections.

Evaluating the arrangement of SIUs has become an important issue in recent years [23–29]. ASTM in the standards of D6433-11 describes a method for evaluating the arrangement of SIUs. This standard considers the allowable error in calculating the PCI and the standard deviation of PCI of inspection units in the section. The arrangement of SIUs is determined by using systematic random sampling [27].

The proposed method using ASTM is generally unable to define the accurate PCI with the calculated number of SIUs which leads to a high network inspection error (NIE). This method considers the selection of additional SIUs for removing this limitation which leads to additional inspection cost and time. The optimized arrangement of SIUs is not achieved by this method.

However, the previous methods do not have any knowledge basis, with a lack of consideration of the PCI in arranging the SIUs with a low PCI spectrum and high sensitivity to length of sections, and thus there is little tradeoff between the number and positions of SIUs. To overcome these limitations, a method for optimizing the arrangement of SIUs should be carried out.

The survey of a pavement section requires various sampling patterns. Therefore, to experiment with these sampling patterns increases the computation time exponentially with the number of inspection units. So, the different computational methods are required for solving the present problem with acceptable time, cost, and inspection errors.

To determine the optimal arrangement of SIUs is to find the optimum sampling pattern with the optimum number of SIUs and the minimum section inspection error (SIE). Some tradeoff between the NIE-inspection time-numbers of SIUs (NSIUs) is a complex problem, which can be a non-deterministic (NP-hard) problem based on computational complexity in the worst case.

Exact algorithms—such as branch and bound algorithms—may achieve global optimum solutions for small problems. However, these algorithms cannot solve NP-hard problems in the desired time and the present problem of a pavement network can include thousands of pavement sections. Another group of algorithms are heuristics. Heuristics produce solutions that are close enough to optimum solutions in an acceptable time. Artificial intelligence (AI) tools have rapidly replaced classical methods in recent decades. Meta-heuristics are a type of AI proposed for achieving near-optimum results faster than previously methods [30, 31]. AI tools are generally used in various civil engineering problems [32].

Recent studies focusing on these issues show that AI and PMS are complementary to each other [31, 33–38].

In the last two decades, several meta-heuristics such as the genetic algorithm (GA) and particle swarm optimization (PSO) have been used for solving different optimization problems. These methods are powerful optimization techniques for solving complex problems [39–41]. PSO and GA have been used by several researchers in pavement management problems in recent years [2, 42–45].

Hybridization of meta-heuristics is commonly done by combining the components of algorithms. There are different types of hybrid meta-heuristics that lead to a variety of solutions [46].

When combining PSO and GA together, we use different conditions such as PSO-GA, GA-PSO, PSO-PSO, GA-GA, PSO in GA, GA in PSO, PSO in PSO, and GA in GA in this chapter.

PSO has a higher speed than GA, although the convergence of GA is better than that of PSO. Therefore, the hybrid approach can consider characteristics of both PSO and GA. The main purpose of this chapter is to develop various states of hybrid PSO and GA for optimally arranging the SIUs and selecting the more precise of them in simulating the inspection process for different constraints in a framework as an expert system. Optimal arrangement of SIUs for the sections of a pavement network leads to tradeoff between the minimum total network error, inspection time, and number of SIUs.

This chapter is organized as follows. The next section gives a brief description of PCI calculation. The data collection of a case study of a pavement network is presented in Sect. 3. The empirical method for determining the arrangement of SIUs is presented in Sect. 4. The importance of the present problem is described in Sect. 4. The definitions and formulations of the problem are presented in Sect. 5. PSO and GA are then used to optimize the arrangement of SIUs in Sects. 6 and 7, respectively. The hybrid PSO and GA arrangers are presented in Sect. 8. Section 9 then presents and discussed the results of the various approaches and, finally, Sect. 10 summarizes the conclusions.

2 Pavement Condition Index (PCI)

The PCI is a numerical rating index, based on a scale from 0 to 100, developed by the collective judgment of experienced pavement engineers to assess the pavement structural integrity, surface operational condition, and required level of Maintenance and Rehabilitation (M&R) in a visual inspection. This inspection collects the type, quantity, and severity of distress [47]. PCI was suggested by the U.S. Army Corps of Engineers for airfield and road pavement and is used all over the world by many agencies [10].

A general procedure for determining PCI of pavement sections is illustrated in Fig. 1 by the inspection units.

An inspection unit is defined as a portion of a pavement section designated only for the purpose of pavement inspection and identify as the shortest unit in the pavement network [10]. Inspection unit size considers a specified area for different pavements and is commonly determined based on the budget for agencies for the considered network.

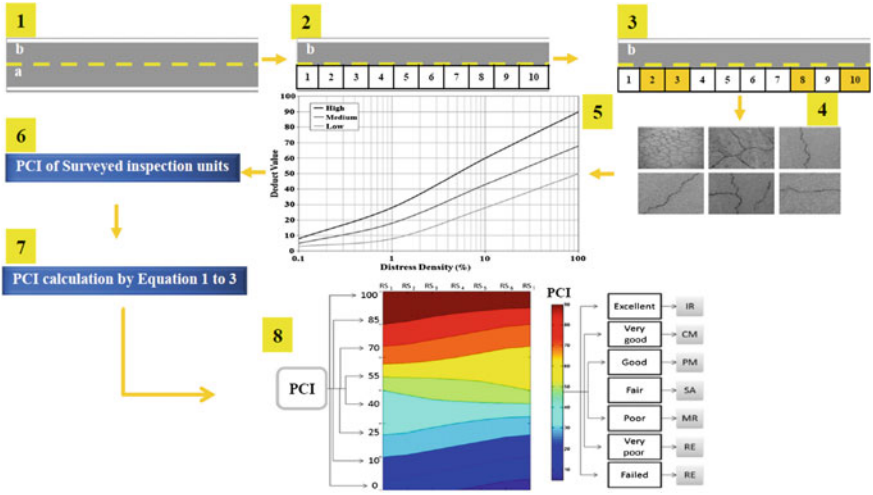

Fig. 1 General framework for calculating the PCI of a section (Moghadas et al. 48)

The PCI of the inspection units is calculated based on measured distress characteristics. The PCI of the sections is determined as the three formulas below:

Formula 1: The sizes of inspection units in the section are not equal: PCI of the section is evaluated by area weighted averaging the PCI of SIUs according to (1)

Formula 2: The additional inspection units are required: the implementation of (2) is required for evaluating the PCI of the section

Formula 3: A particular state of Formula 1 with an equal size of inspection units: PCI is calculated by averaging the PCI of SIUs as in (3)

$$\mathrm{PCI}_s = \frac{\sum_{i=1}^{n} \mathrm{PCI}_{\mathrm{siu}} \times A_{\mathrm{siu}}}{\sum_{i=1}^{n} A_{\mathrm{siu}}} \tag{1}$$

$$\mathrm{PCI}_{sa} = \frac{\mathrm{PCI}_s\left(A_s - \sum_{i=1}^{A} A_{\mathrm{add}}\right) + \mathrm{PCI}_{\mathrm{add}} \times \sum_{i=1}^{A} A_{\mathrm{add}}}{A_s} \tag{2}$$

$$\mathrm{PCI}_s = \frac{\sum_{i=1}^{n} \mathrm{PCI}_{\mathrm{siu}}}{n} \tag{3}$$

where PCI_s is the PCI of section, $\mathrm{PCI}_{\mathrm{siu}}$ is the PCI of each SIU, and A_{siu} is the area of each SIU. In addition, n is the number of SIUs in the section, $\mathrm{PCI}_{\mathrm{sa}}$ is the PCI of section with additional SIUs, and A_s is the total area of section. Furthermore, $\mathrm{PCI}_{\mathrm{add}}$ is the area weighted average PCI of additional SIUs, A_{add} is the area of additional SIUs, and A is the area of additional SIUs in the section. The PCI is used by researchers in some existing works [2, 12, 17, 19, 21, 22].

3 Data Collection

For the present research, an infield pavement survey by the inspectors is conducted for data collection. This work takes about a month for a pavement network located in district No. 16 of Tehran municipality as a case study by four inspection teams. The characteristics of the various distresses are written in the "infield survey forms" and the place of distresses identified by a GPS devices (Fig. 2). A GARMIN eTrex® 30 GPS is the GPS device used for collecting the data in-field by the surveyors.

This network is 294,851 m long and is divided into 3,925 sections based on the average daily traffic (ADT), construction, and maintenance history, pavement width, drainage conditions, and road type. Therefore, each 5-m unit is considered for increased inspection accuracy and then provides an accurate database.

For simplicity of presentation, the PCI rating scale may be presented in three categories including good (PCI = 71–100), fair (PCI = 41–70), and poor (PCI = 0–40) with 89, 10 and 1 % normally used for illustrative results, respectively.

Fig. 2 Studied pavement network with locations of distress

4 Empirical Method

The American Society for Testing and Materials (ASTM) presents a model for estimating the SIUs by considering factors such as the allowable error in the PCI calculation, standard deviation of the PCI inspection units in the section, and total number of inspection units. This method is an empirical method for determining the number and placement of SIUs.

In this method, the arrangement of SIUs is determined by using systematic random sampling. Therefore, the number of SIUs is generally unable to define an accurate PCI for the section and leads to a high inspection error for the sections in a pavement network. For solving this challenge it is necessary to select additional SIUs that can lead to additional cost and time.

This method has a procedure for determining the number and place of SIUs as follows [27]:

- Identify branches of the pavement network with different uses such as roadways and parking.
- Divide each branch into sections based on the pavements design, construction history, traffic, and condition.
- Divide the pavement sections into sample units (inspection units).
- It is necessary to be able to relocate accurately the inspection units to allow verification of current distress data, to examine changes in condition with time of a particular inspection unit, and to enable future inspections of the same inspection unit if desired.
- Select the inspection units to be surveyed as SIUs. The number of SIUs may vary from the following: all of the sample units in the section, a number of sample units that provides a 95 % confidence level, or a lesser number.
- The minimum number of SIUs (n) within a given section to obtain a statistically adequate estimate (95 % confidence) of the PCI of the section is calculated using (4) and rounding n to the next highest whole number:

$$n = \frac{N \cdot s^2}{\frac{E^2}{4}(N-1) + s^2} \tag{4}$$

where E is the acceptable error in estimating the section PCI and is commonly equal to ±5 PCI points, and s is the standard deviation of the PCI from one sample unit to another within the section. When performing the initial inspection, the standard deviation is assumed to be 10 for asphalt pavements. This assumption should be checked as described below after PCI values are determined. For subsequent inspections, the standard deviation from the preceding inspection should be used to determine n. Here, N is the total number of sample units in the section.

- If obtaining the 95 % confidence level is critical, the adequacy of the number of SIUs must be confirmed. The number of SIUs was estimated based on an assumed standard deviation. The actual standard deviation(s) can be calculated using (5):

$$s = \left(\sum_{i=1}^{n} (\mathrm{PCI}_i - \mathrm{PCI}_s)^2 / (n-1) \right)^{\frac{1}{2}} \tag{5}$$

where PCI_i is the PCI of SIUi, PCI_s is the PCI of the section, and n is the total number of SIUs.

- Calculate the revised minimum number of SIUs using (4) and the calculated standard deviation using (5). If the revised number of SIUs is greater than the number of inspection units already surveyed, select and survey additional random inspection units. These inspection units should be spaced evenly across the section.
- Repeat the process of checking the revised number of SIUs and surveying additional random inspection units until the total number of SIUs equals or exceeds the minimum required sample units (n) in (4), using the actual total sample standard deviation.
- Once the number of SIUs has been determined, compute the spacing interval of the units using systematic random sampling. The SIUs are spaced equally throughout the section with the first sample selected at random. The spacing interval (i) of the SIUs is calculated using (6):

$$i = \frac{N}{n} \tag{6}$$

The first SIU is selected at random from inspection units 1 through i. The inspection units within a section that are successive increments of the interval i after the first randomly selected unit are also inspected. To clarify this method, consider a section with 20 inspection units (Fig. 3). This method conducts several tries for achieving an acceptable PCI and to identify sampling states that have various arrangements of SIUs. Trials of the ASTM method for the section is shown in Table 1.

Table 1 shows that this method is required because of the extreme length of time taken for analyzing pavement network with thousands of sections.

100	86	71	16	49	97	55	24	83	9	64	99	43	27	15	76	32	19	95	63

Fig. 3 First sampling state of a pavement section using the ASTM method

Table 1 Calculation of section PCI using the ASTM method

Sampling states	Exact PCI	Number of SIUs	Additional number of SIUs	Obtained PCI	SE (%)
Try 1	56.15	4	0	39.25	16.9
Try 2		6	2	56.667	0.517
Try 3		8	4	56.50	0.350
Try 4		12	8	55.833	0.317
Try 5		9	5	56.444	0.294
Try 6		12	8	56.417	0.267
Try 7		11	7	56.273	0.123
Try 8		10	6	56.20	0.050
Try 9		17	13	56.1176	0.0324

5 Problem Definitions

With regard to the importance of determining the accurate pavement condition in the pavement inspection process, the arrangement of SIUs is one of the most effective variables. As the previous methods presented all over the world such as the ASTM method (in Sect. 4) for arrangement of SIUs had limitations that finally led to high SE in a field survey, it is necessary to develop intelligent methods for optimizing the arrangement of SIUs in the sections of a pavement network. To remove these limitations, two objective functions are considered in the proposed methods which can be minimized in the error of the inspection process as follows.

5.1 Sampling Error

The inspection error or sampling error (SE) of each pavement section is evaluated by a definition of error. In this chapter, SE is the absolute value of the difference between the real condition and condition obtained from the proposed method. For a pavement network with N sections, each of which includes n inspection units, surveying all of the inspection units in field sampling leads to the exact PCI. However, for the proposed method, it is required to select m inspection units as SIUs. For the main purpose of this research, the objective function for sections is formulated mathematically as (7). A comparison between the routine and optimum inspection is shown in Fig. 2.

$$\min E_s = \left(\frac{1}{n} \sum_{i=1}^{n} \mathrm{PCI}_{si} - \frac{1}{m} \sum_{j=1}^{m} \mathrm{PCI}_{sj} \right) \tag{7}$$

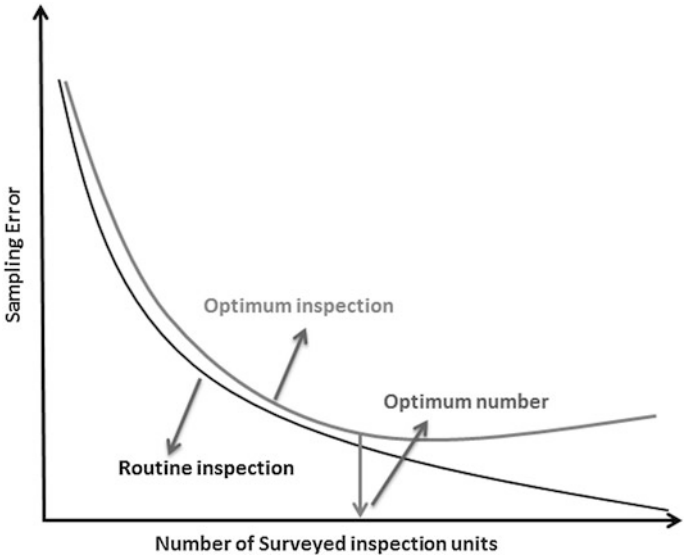

Fig. 4 Comparison between routine and optimum inspection

Subject to $1 \leq m \leq n$ where E_s is the SE for each section, n is the total number of inspection units in real sampling considered equal to the number of SIUs, and m is the number of SIUs based on the proposed method. In addition, PCI_{si} is the PCI of the ith SIU in the sth section for real sampling and PCI_{sj} is the PCI of the jth SIU in the sth section based on the proposed method. The constraint for this objective function is sample m of SIUs, which is at least equal to one SIU and the maximum equal to all of the inspection units. Figure 4 illustrates that, by increasing the number of SIUs, the SE is decreased in the routine inspection whereas the optimum inspection has the optimum number of SIUs by surveying a specific number of SIUs.

5.2 Network Inspection Error (NIE)

NIE is an error in evaluating the total SE of sections in a pavement network. For evaluating the error in the inspection of pavement network as TNIE it is necessary to determine the variance of the SE of total sections because the variance is one of the functions for evaluating the accuracy of methods. However, the objective function for NIE is formulated mathematically as (8) which is applied as the main function for the proposed algorithms:

$$\min \mathrm{Var}(E_s) = \frac{1}{N}\sum_{s=1}^{N}\left(E_s - \overline{E}\right) = \frac{1}{N}\sum_{s=1}^{N}\left(\left(\frac{1}{n}\sum_{i=1}^{n}\mathrm{PCI}_{si} - \frac{1}{m}\sum_{j=1}^{m}\mathrm{PCI}_{sj}\right) - \frac{E_T}{N}\right) \tag{8}$$

where

$$E_T = \sum_{s=1}^{N} E_s = \sum_{s=1}^{N}\left(\frac{1}{n}\sum_{i=1}^{n}\mathrm{PCI}_{si} - \frac{1}{m}\sum_{j=1}^{m}\mathrm{PCI}_{sj}\right) \tag{9}$$

$$\overline{E} = \frac{E_T}{N} \tag{10}$$

subject to

$$\mathrm{Var}(E_s) = \frac{1}{N}\sum_{s=1}^{N}\left(E_s - \overline{E}\right) \tag{11}$$

$$\text{Total Number of SIUs} = \sum_{s=1}^{N}(m_s) \tag{12}$$

$$T_t = \sum_{i=1}^{N}(T_i) + \sum_{p=1}^{N}(T_p) + \sum_{o=1}^{N}(T_o) \tag{13}$$

where E_T is the summation of sections SE, N is the number of total sections in a network, and Var(E_s) is variance of SE, namely total network error. In addition, m_s is the total number of SIUs. The constraints for this objective function as three fitness values are the total network error, the number of SIUs. Furthermore, T_t is the total computational time (CPU time), including T_i as the input time, T_p as the process time, and T_o as the output time.

However, based on recent studies, especially on the ASTM method, the arrangement of SIUs is determined by the number of SIUs and using systematic random sampling for their placement in the pavement sections.

So, the challenges lead to limitations in the application of the related methods. In this regard, the inspection process must be implemented by intelligent methods for eliminating these challenges. In this research, the various approaches of hybrid PSO and GA are applied for optimizing the arrangement of SIUs in a pavement network.

6 PSO Arranger

PSO is an optimization method for solving various complex problems. For the present problem of this research, PSO is an arranger of the SIUs in pavement sections. The basic idea of the PSO mimics swarm behavior of birds or fishes and the relationship between them. PSO has several points in the search space as particles in a swarm for updating the current condition of the individuals [49].

PSO simulates the movements of these particles and leads to the exploration of various regions in the search space for the global optima. Unlike GA, PSO has no evolutionary operators [50]. In the PSO arranger, the inspection process is simulated for the sections of a pavement network. In this arrangement, the sampling states are simulated as the particles and the binary coding is the section sampling. Particles have velocities to move their positions in the search space and memory that retains its previous best position as personal best (pbest) for remembering the best achieved position of the search space [49]. The particles accelerate toward their pbest and the direction of movement is toward the best particle of a topological neighborhood. PSO arranger keeps the best value and position in particles of the swarm which is called gbest. Particles move toward its best previous position and toward the best particle in the swarm [50]. To ensure convergence, a parameter has to be applied carefully and this parameter is the constriction factor [51]. Therefore, the velocity and position are calculated from (9) and (10), respectively.

$$\min \mathrm{Var}(E_s) = \frac{1}{N}\sum_{s=1}^{N}\left(E_s - \overline{E}\right) = \frac{1}{N}\sum_{s=1}^{N}\left(\left(\frac{1}{n}\sum_{i=1}^{n} PCI_{si} - \frac{1}{m}\sum_{j=1}^{m} PCI_{sj}\right) - \frac{E_T}{N}\right) \tag{14}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{15}$$

where c_1, c_2 are two positive constants that are called learning factors, including cognitive and social learning parameters, respectively; r_1, r_2 are uniformly disturbed random numbers within 0 and 1. v_{id} and x_{id} are the velocity and position of ith particle in dimension d, respectively. The constriction factor defined in [52] can be calculated from

$$\chi = \frac{2}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|} \tag{16}$$

Experience has shown that the constriction factor needs a boundary for controlling the solution space and facilitating better convergence. A boundary condition is *reflecting walls* used with the PSO arranger. In this boundary, when a particle contacts the walls, the sign of the velocity is reversed and the particle moves back toward the solution space [53]. In addition to continuous PSO, binary PSO was first proposed by Kennedy and Eberhart [54]. In binary PSO, the position

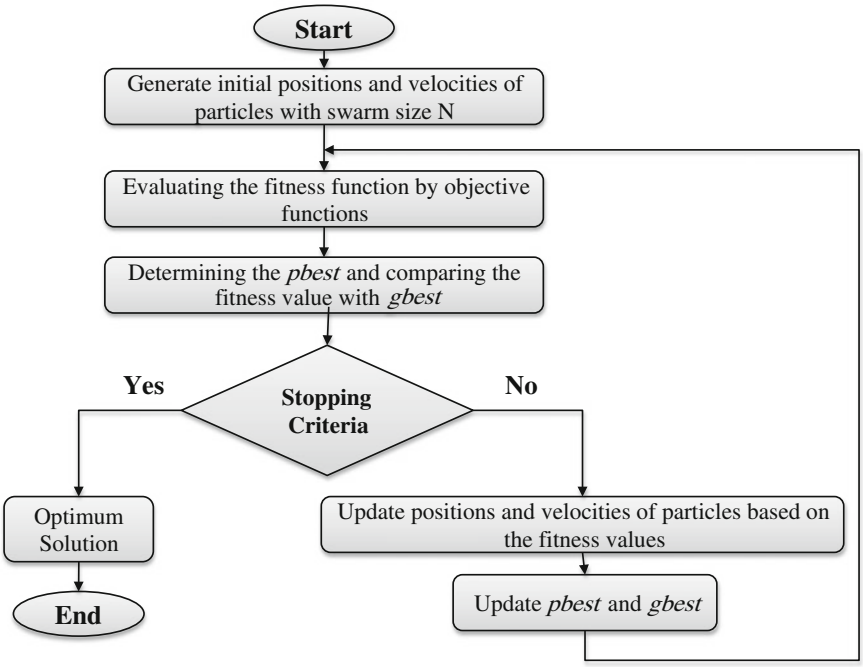

Fig. 5 Flowchart of the PSO arranger

of particles is defined by a binary vector. The velocity vector is associated with the probability of each dimension taking the value 1. A sigmoid function is considered for changing the velocity from values 0 to 1 and conversely. r is compared with the value of the sigmoid function in (12) and then the position updated using (13).

$$S(v_{id}) = \frac{1}{1 + \exp(-v_{id})} \tag{17}$$

$$x_{id} = \begin{cases} 1, & r < S(v_{id}) \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where r is a random uniformly number generated within 0 and 1.

The general flowchart of the PSO arranger is shown in Fig. 5.

7 GA Arranger

In this research, a GA is considered as the arranger of SIUs in the sections of pavement network. The first GA proposed by Holland and based on Darwin's evolution law became a strong tool for solving complex problems [41]. Each GA

has three main operators—selection, crossover, and mutation. In this research a method is considered for each operator that can be used to arrange the SIUs in the pavement sections.

7.1 Selection

The selection operator is important in GA for selecting individuals because this operator has a significant effect on solutions convergence [55]. The selection operator is implemented for selecting chromosomes to generate a new population after crossover.

Tournament selection is a selection operator method which involves running several "tournaments" among a set of chromosomes chosen at random from the population. The winner of each tournament (the one with the best fitness) is chosen for crossover operation. This operator has a parameter, namely selection pressure, which it uses to select the better individuals based on fitness value with more chances and it can be adjusted by changing the tournament size. If the tournament size is lower, stronger individuals have a higher chance of being selected (Fig. 6a). Tournaments are repeated until the mating pool for generating new offspring is filled [56].

7.2 Mutation

Mutation is an operator for mutating the gene(s) in the parents and making additional population for achieving the optimum solution. A mutation operator can be a

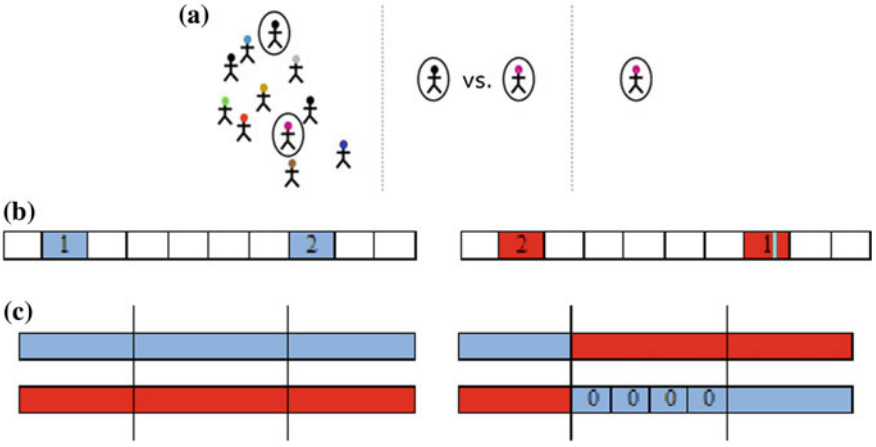

Fig. 6 GA arranger operators. **a** Tournament selection. **b** Swap mutation. **c** Dissociated crossover

swap mutation or an interchanging mutation. In this method, two genes are chosen randomly and interchange the related values [57]. This method is shown in Fig. 6b.

7.3 *Crossover*

Dissociated crossover is a method that randomly selects two cut points in both parents and divides each chromosome into three parts (Fig. 6c). The first part is the genes copied in the related offspring before cut points are exchanged. The second part is those exchanged between two parents with the genes of second offspring equal to zero. The third part is from an exchange in the related part of the parent [57].

The GA arranger implementation cycle is as shown in Fig. 7.

8 Hybrid PSO and GA Arrangers

Hybridization of the optimization methods is an attempt to use the advantages of a combination of methods for better results in convergence and exploration. This work overcomes the limitations of each individual method. The hybridization of

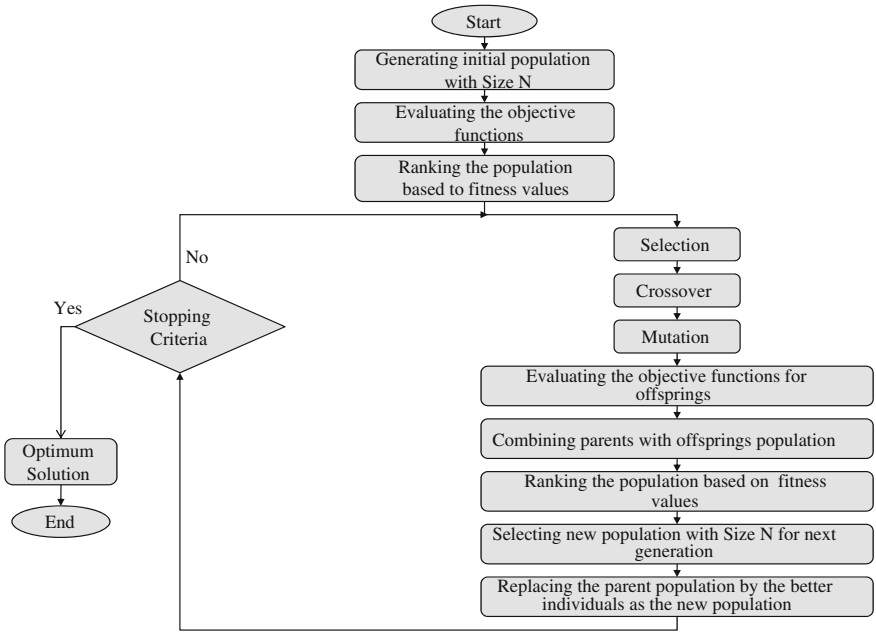

Fig. 7 GA arranger flowchart

PSO and GA is a commonly used hybrid optimization method [58–64]. In this chapter, various approaches of hybrid PSO and GA are considered for optimal arrangement of SIUs in a pavement network. These approaches are as follows.

8.1 PSO-GA Arranger

This type of hybridization starts with parameters and relationships of the PSO arranger, and then continues by GA arranger to achieve the optimal solution. This method follows the procedure as in Fig. 8.

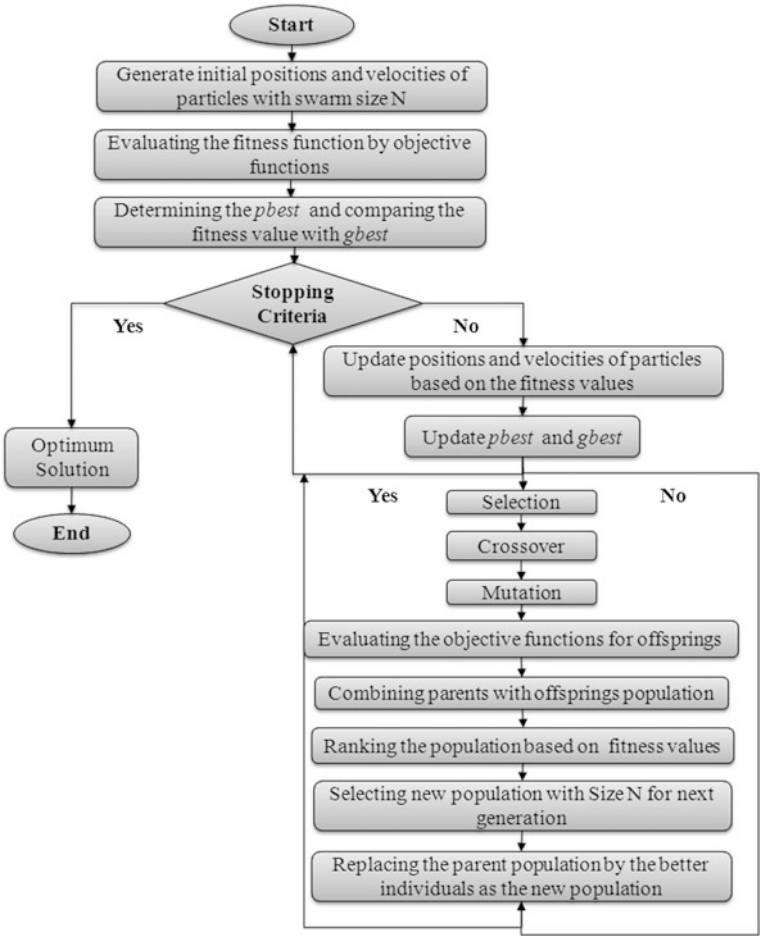

Fig. 8 PSO-GA arranger procedure

8.2 GA-PSO Arranger

This hybrid method is the same as the PSO-GA arranger but exchanges the PSO and GA in the implementation procedure.

8.3 PSO-PSO Arranger

In the PSO-PSO arranger, the PSO arranger is implemented twice.

8.4 GA-GA Arranger

In the GA-GA arranger, the GA arranger is implemented twice.

8.5 GA in PSO Arranger

In this method, the PSO arranger begins to determine the pbest and gbest and then the GA runs by the related operators. The algorithm continues to update the pbest and gbest or achieve the optimal solution. The procedure for this method is illustrated in Fig. 9.

8.6 PSO in GA Arranger

As with the GA in PSO arranger, the PSO in GA arranger starts with the GA arranger and continues until parameters and relationships of PSO appear.

8.7 GA in GA Arranger

The GA in GA arranger is a hybridization approach whereby two GA arrangers operate together. It work by putting the parameters and relationships from a GA into another GA arranger.

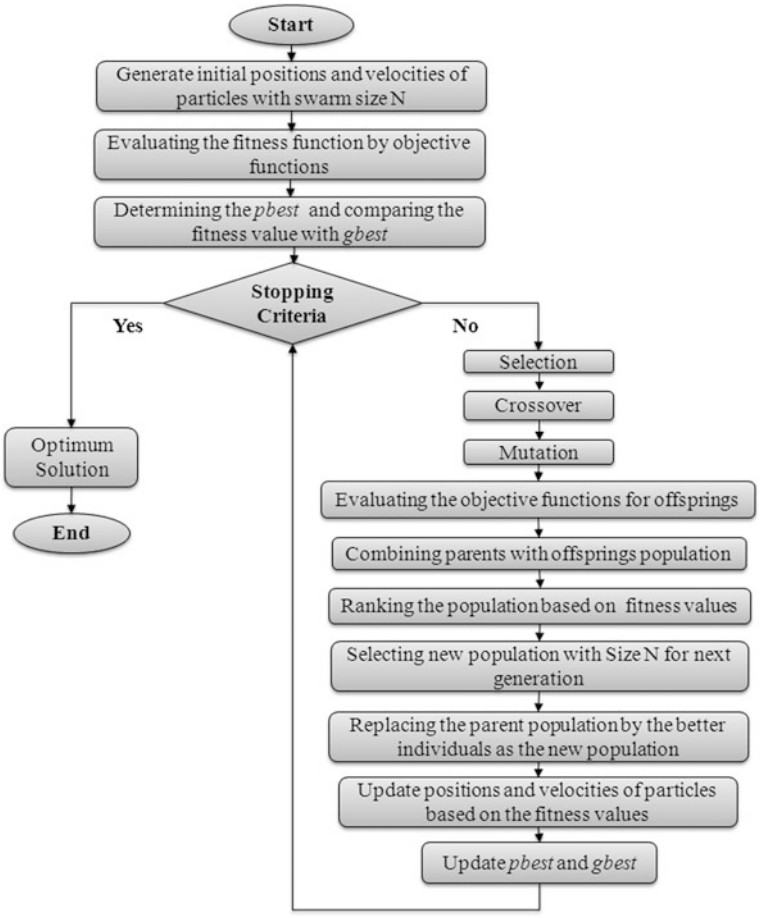

Fig. 9 GA in PSO arranger procedure

8.8 PSO in PSO Arranger

This is the same as the GA in GA arranger; the PSO in PSO arranger is another hybridization approach whereby two PSO arranger are put together. This works by putting parameters and relationships from a PSO into another PSO arranger.

9 Results and Discussion

In the present work, the hybrid methods of GA and PSO were coded and have been implemented in MATLAB 7.12.0 on a PC with Intel(R) Core i3 3.30 GHz CPU, 8 GB RAM, and Windows 7 operating system. These methods require parameters

Table 2 Best parameters for the methods

Method	Parameters							
	PZ	MN	CR	MR	TS	CF	(c_1)	(c_2)
PSO-GA	20	200	0.1	0.05	2	0.7298	1.4962	1.4962
GA-PSO	15	200	0.1	0.05	2	0.7298	1.4962	1.4962
PSO-PSO	5	200	–	–	–	0.7298	1.4962	1.4962
GA-GA	75	200	0.1	0.01	2	–	–	–
PSO in GA	70	200	0.1	0.01	2	0.7298	1.4962	1.4962
GA in PSO	15	200	0.1	0.01	2	0.7298	1.4962	1.4962
PSO in PSO	15	200	–	–	–	0.7298	1.4962	1.4962
GA in GA	15	200	0.1	0.05	2	–	–	–
GA	80	200	0.1	0.01	2	–	–	–
PSO	10	200	–	–	–	0.7298	1.4962	1.4962

Note The values of constriction factors c_1 and c_2 are considered by Clerc and Kennedy [40]
PZ Population or swarm size
MN Maximum number of generations
CR Crossover rate
MR Mutation rate
TS Tournament size
CF Constriction factor
c_1 Cognitive learning factor
c_2 Social learning factor

to be set properly in good implementations. Based on this issue, the parameters of the various methods are presented in Table 2. The results of this research involve two parts, network and project level, and evaluation of the sampling SE, number of SIUs, implementation time (CPU time), convergence diagrams, PCI, and NIE for proposed methods.

9.1 Network Level

For this level of study, the pavement networks with all of the sections analyze and compare together. In this part, a comparison is made based on the minimal NIE equal to 0.0001 and it leads to a number of SIUs and CPU time. For this work, ten runs for each proposed method are made and the values of average, minimum, and maximum recorded. These results are illustrated in Table 3 and 4 and Figs. 10 and 11. Figure 10 illustrates a comparison between all of the methods and Fig. 11 shows a comparison between the hybrid methods.

Table 3 illustrates that PSO-GA has better results than others based on the average values in the ten iterative runs. In addition, GA and PSO alone, GA in GA, and GA-GA are weaker than the others in terms of CPU time and number of SIUs. Based on Fig. 10, GA has higher values of CPU time and number of SIUs than the

Table 3 Comparison of hybrid methods in optimizing arrangement of SIUs

Method	PSO-GA		GA-PSO		PSO-PSO		GA-GA		PSO in GA	
Runs	Num.	Time	Num.	Time	Num.	Time	Num.	Time	Num.	Time
1	19,930	339.60	20,137	413.96	20,080	227.97	20,471	1944.13	20,102	339.22
2	19,958	324.86	19,950	287.06	20,117	225.48	20,253	1685.91	20,145	388.65
3	20,197	361.42	20,270	386.75	20,080	242.68	20,671	1272.69	20,121	427.44
4	20,015	330.19	20,230	373.27	20,117	239.52	20,595	1154.55	20,094	392.94
5	20,197	366.21	20,014	362.40	20,215	283.10	20,421	1547.09	20,145	382.28
6	20,143	448.13	20,052	373.99	20,080	240.74	20,671	1219.98	20,050	522.83
7	20,083	349.78	20,070	360.05	20,117	237.93	20,671	1229.35	20,109	323.18
8	20,006	409.83	20,158	414.90	20,215	281.00	20,671	1260.45	20,104	390.26
9	20071	410.82	19,950	298.34	20,160	257.34	20,671	1228.30	19,957	520.68
10	20,015	326.91	20,270	401.25	20,214	300.48	20,671	1250.17	20,099	325.68
Average	20,061	366.77	20,110	367.20	20,139	253.62	20,576	1379.26	20,092	401.32
Std.	93.88	42.36	122.03	43.87	57.26	25.84	146.78	259.18	54.80	71.57
Min	19,930	324.86	19,950	287.06	20,080	225.48	20,253	1154.55	19,957	323.18
Max	20,197	448.13	20,270	414.90	20,215	300.48	20,671	1944.13	20,145	522.83

Note Num. is the number of SIUs, *Time* is the implementation CPU time of methods and, *Std*. is standard deviation

Table 4 Comparison of hybrid methods, PSO and GA in optimizing arrangement of SIUs

Method	GA in PSO		PSO in PSO		GA in GA		GA		PSO	
Runs	Num.	Time	Num.	Time	Num.	Time	Num.	Time	Num.	Time
1	20,049	314.22	20,169	559.50	20,406	1875.55	20,524	1814.742	20,152	298.36
2	19,951	363.65	20,115	539.52	20,510	1900.99	20,493	1958.018	20,152	299.09
3	20,053	402.44	20,236	672.82	20,401	1822.67	20,522	19231.22	20,152	313.56
4	20,037	367.94	20,115	529.90	20,486	1842.54	20,565	18664.55	20,284	368.03
5	20,132	357.28	20,236	669.85	20,511	1869.77	20,499	18245.47	20,102	346.21
6	20,250	497.83	20,122	498.70	20,531	1879.66	20,596	18549.56	20,391	322.96
7	20,199	298.18	20,164	562.02	20,495	1868.64	20,566	18762.98	20,327	348.83
8	20,204	365.26	20,115	497.20	20,454	1895.47	20,520	18961.65	20,152	302.34
9	20,250	495.68	20,236	618.98	20,412	1901.65	20,558	18645.68	20,284	364.06
10	20,199	300.68	20,122	494.98	20,439	1906.54	20,577	19548.64	20,102	339.65
Average	20,132	376.32	20,163	564.35	20,464	1876.35	20,542	15438.25	20,210	330.31
Std.	103.78	71.57	54.06	67.78	48.33	27.21	34.90	7151.72	102.30	26.58
Min	19,951	298.18	20,115	494.98	20,401	1822.67	20,493	1814.74	20,102.00	298.36
Max	20,250	497.83	20,236	672.82	20,531	1906.54	20,596	19548.64	20,391.00	368.03

Note Num. is the number of SIUs, *Time* is the implementation CPU time of methods, and *Std*. is standard deviation

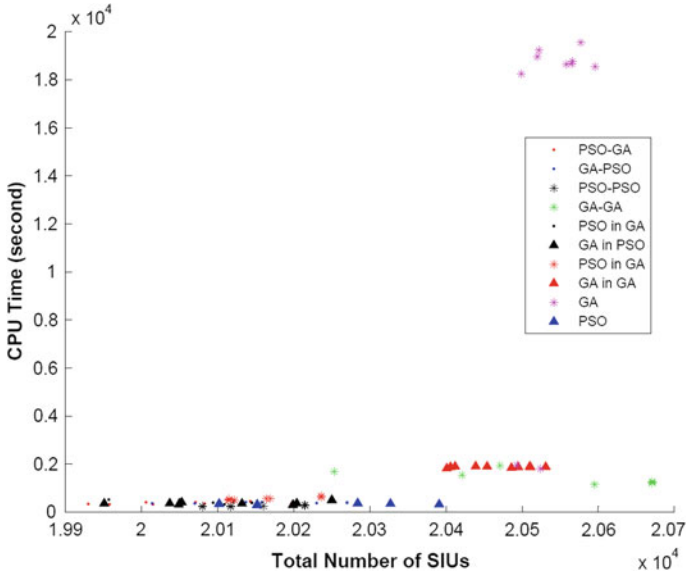

Fig. 10 Comparison of the proposed methods in CPU time versus number of SIUs in the network level

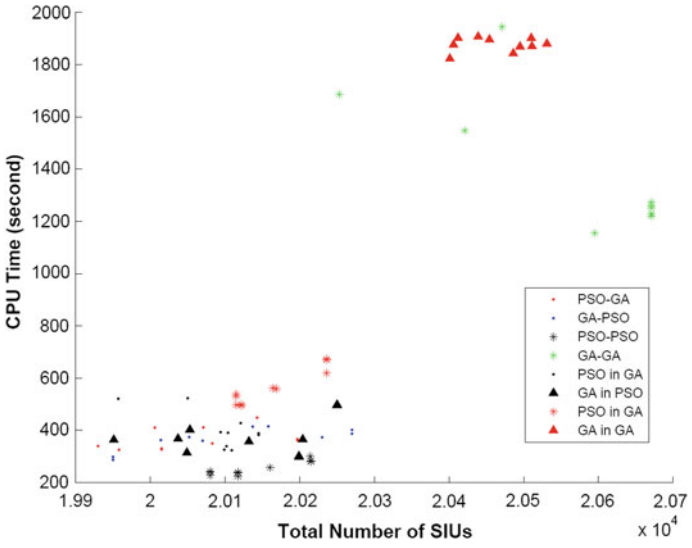

Fig. 11 Comparison between the proposed hybrid methods in CPU time versus number of SIUs in the network level

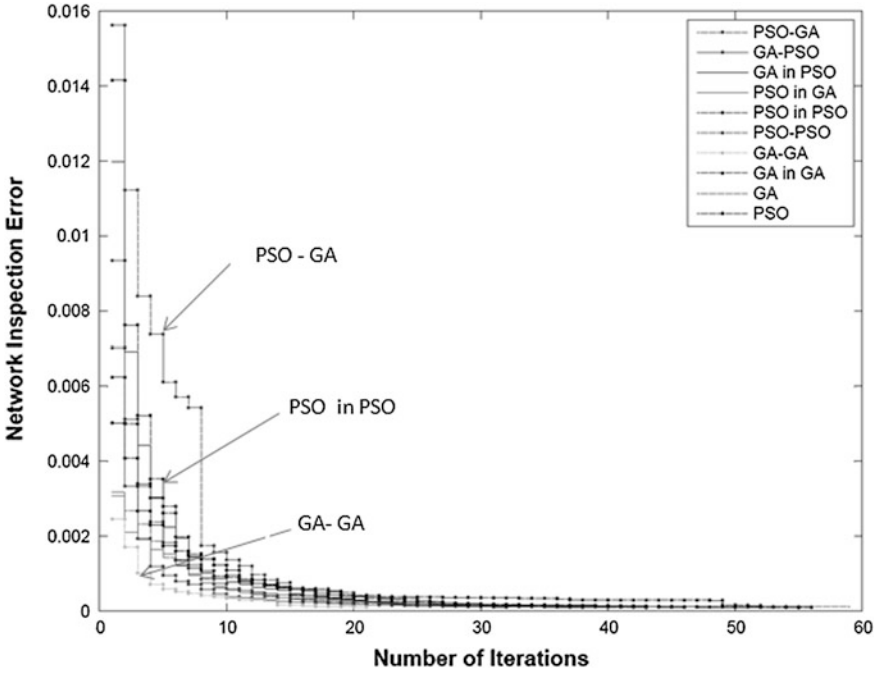

Fig. 12 Comparison of the various proposed methods in convergence to optimal solution

other methods and the other methods have good results. Figure 11 illustrates that GA-GA and GA in GA work more weakly than other hybrid methods for the problem of this research at the network level. PSO-GA and GA-PSO have good solution limits. In addition to the above cases, the convergence to optimal solution is an important issue in the optimization problems and a comparison between the proposed methods is shown in Fig. 12.

Figure 12 illustrates that all the proposed methods have obtained good converged solutions based on NIE versus iterations. However, PSO in PSO and PSO-PSO are weaker than other methods, although PSO-GA and GA-PSO have more acceptable convergence.

9.2 Project Level

Part of the studied pavement network is considered for studying the results and comparing the effects of proposed methods at project level (see Fig. 13). The results are implemented and presented in Arc Map software as Figs. 14 and 15. Figure 14 illustrates the PCI of sections for the part of pavement network at the project level

Fig. 13 Pavement network of district No. 16 of Tehran municipality with the considered part

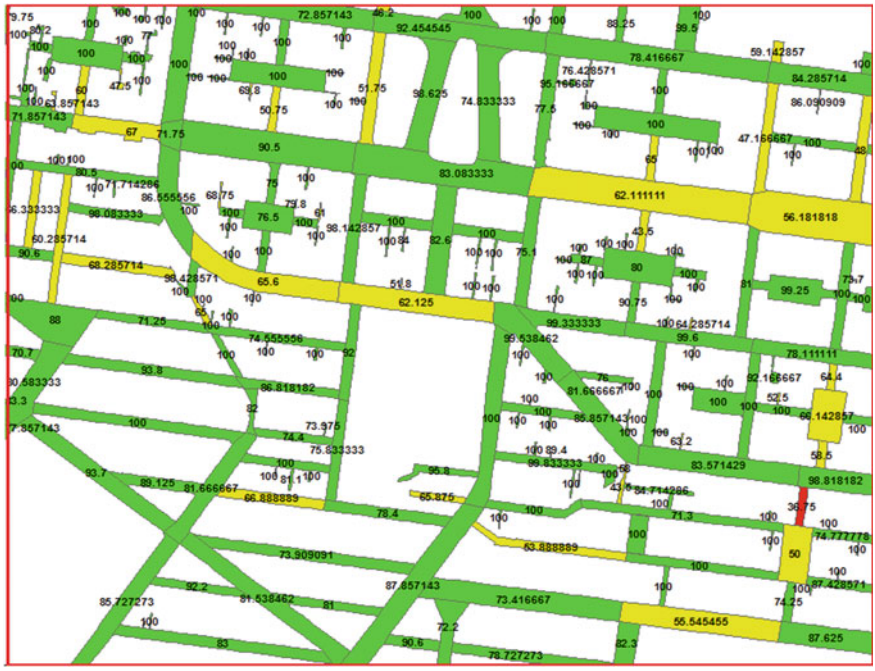

Fig. 14 PCIs of a considered part of the studied pavement network with proposed methods at project level (PSO-GA)

Fig. 15 SEs of a considered part of the studied pavement network with proposed method at project level

sampled by the proposed methods and Fig. 15 shows the related SE of sections for this part. This figure is equal to the differences between Figs. 13 and 14. Figure 15 illustrates that the SE of all of the methods is acceptable for the sections of a pavement network. The results of the considered part of the studied pavement network is presented and summarized in Fig. 16 for comparing the effect of proposed methods.

Figure 16 includes the SE and number of SIUs (NSIUs) with a comparison between the proposed methods by these parameters. This figure illustrates that all of the methods have acceptable SEs, and that the PSO-GA, GA-PSO, GA in PSO, and PSO in PSO sampled lower NSIUs than other proposed methods.

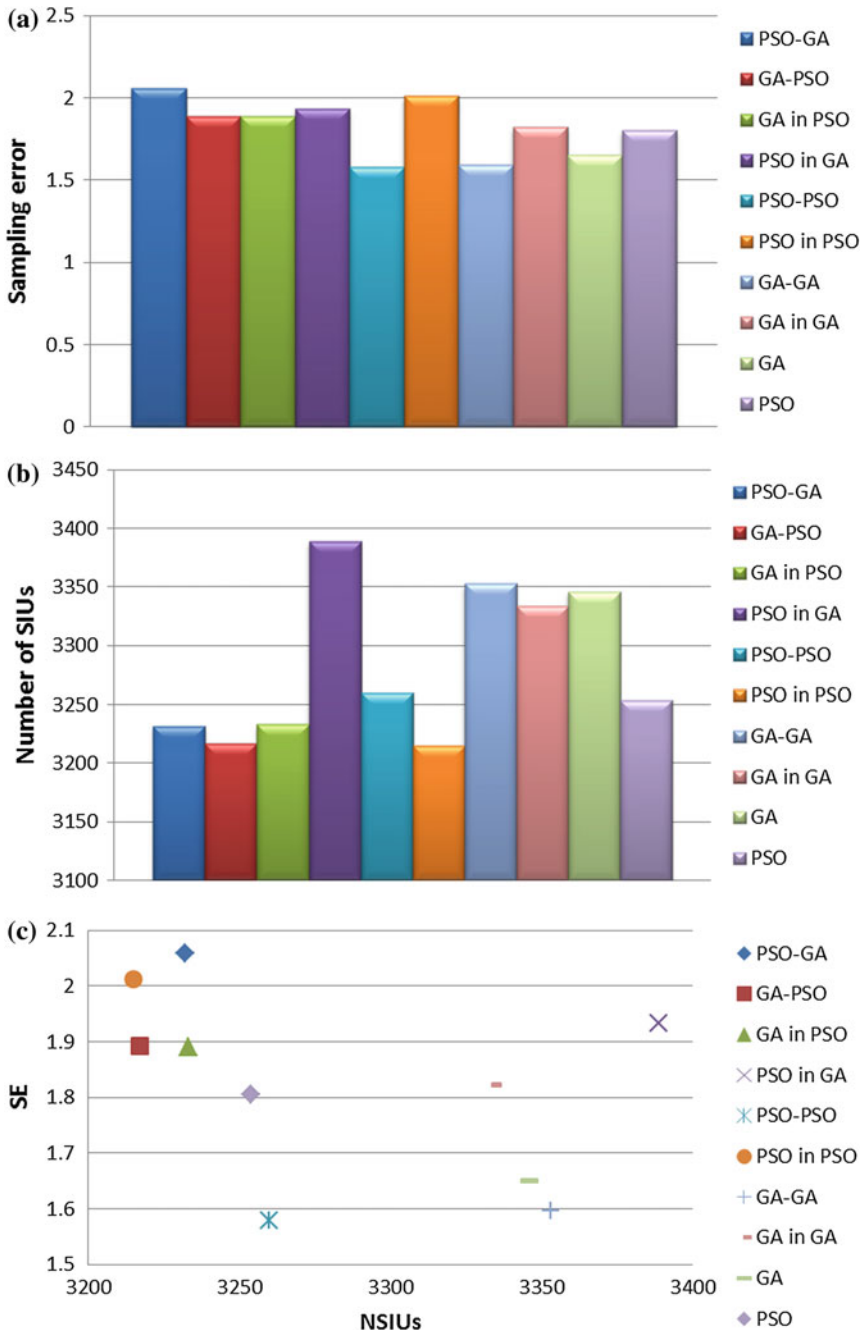

Fig. 16 Comparison between the proposed methods in SE of the considered part of studied pavement network. **a** SE in the considered part of project level. **b** Number of SIUs in the considered part of project level. **c** SE versus number of SIUs in the considered part

10 Conclusions

In this chapter, eight hybrid methods of GA and PSO are proposed with various approaches derived for optimal arrangement of the SIUs with accurate PCI in a comprehensive case study of the pavement network in two levels (network and project levels). This problem is formulated with objectives, including SE and NIE, and is subject to a number of constraints for optimal arrangement of SIUs in the pavement sections of a network.

Finally, these hybrid methods are compared with the PSO and GA alone based on various cases such as estimation accuracy of PCI, SE, convergence diagrams, TNIE, CPU time, and total number of SIUs.

The major findings of this research are as follows:

- The hybrid methods of PSO and GA have been successful for solving the present problem and achieved the least SE in the sections of pavement network with optimal arrangement of SIUs. These approaches are better than PSO and GA.
- Selecting the best approach and the comprehensive sampling procedure is accomplished for obtaining better pavement network analysis. Based on this issue, PSO-GA is the best approach in the network level that generates better results in TNIE, CPU time, and total number of SIUs than other methods in iterative runs.
- The proposed methodology of this chapter can help pavement managers and inspectors use optimal decision making in inspection processes and sampling plans with high accuracy, low analysis time, and low number of SIUs.
- The hybrid methods enable us to obviate the analyzing and planning problems of the inspection process in the pavement network effectively.
- The results show that the proposed methods as well as the sampling process can be simulated and they work for each number of inspection units in the sections and each number of sections in a pavement network better than GA and PSO separately.
- The proposed hybrid methods have been successful for converging to optimal solution and work better than singularly.
- One of the challenges of the methods presented in the literature was limitation in inspection unit PCIs spectrum in the sections. The proposed methods are shown to work for each PCI spectrum and have removed this limitation.
- The reliability of results of hybrid methods of PSO-GA and GA-PSO in the iterative runs is better than other methods based on the NIE, total number of SIUs, and CPU time.
- As the PSO has characteristics such as the simplicity and lack of evolutionary operators on the one hand and good convergence of GA with more exploration of solutions by related operators on the other, hybrid approaches of PSO and GA are useful for increasing the potential of solving the present problem. This issue is shown in the results of this research. The convergence of PSO-GA and GA-PSO is better than the other methods.

References

1. Medury, A., Madanat, S.: Simultaneous network optimization approach for pavement management systems. J. Infrastruct. Syst. **20**(3), 04014010 (2014)
2. Shahnazari, H., Tutunchian, M.A., Mashayekhi, M., Amini, A.A.: Application of soft computing for prediction of pavement condition index. J. Transp. Eng. **138**(12), 1495–1506 (2012)
3. Zhou, G., Wang, L.: Co-location decision tree for enhancing decision-making of pavement maintenance and rehabilitation. J. Transp. Res. Part C **21**, 287–305 (2012)
4. Bianchini, A.: Fuzzy representation of pavement condition for efficient pavement management. J. Comput. Aid. Civil Infra. Eng. **27**, 608–619 (2012)
5. Meegoda, J., Gao, S.: Roughness progression model for asphalt pavements using long-term pavement performance data. J. Transp. Eng. **140**(8), 04014037 (2014)
6. Bogus, S., Song, J., Waggerman, R., Lenke, L.: Rank correlation method for evaluating manual pavement distress data variability. J. Infrastruct. Syst. **16**(1), 66–72 (2010)
7. Su, Y., Kang, S., Chang, J., Hsieh, S.: Dual-light inspection method for automatic pavement surveys. J. Comput. Civ. Eng. **27**(5), 534–543 (2013)
8. Smilowitz, K., Madanat, S.: Optimal inspection and maintenance policies for infrastructure networks. Comput. Aided Civil Infrastruct. Eng. **15**(1), 5–13 (2000)
9. Koch, C., Jog, G., Brilakis, I.: Automated pothole distress assessment using asphalt pavement video data. J. Comput. Civ. Eng. **27**(4), 370–378 (2013)
10. Shahin, M.Y.: Pavement Management for Airports, Roads and Parking Lots, 2nd edn. Springer, New York (2005)
11. Hegazy, T., Elhakeem, A., Ahluwalia, S.S., Attalla, M.: MOST-FIT: support techniques for inspection and life cycle optimization in building asset management. Comput. Aided Civil Infrastruct. Eng. **27**, 130–142 (2012)
12. Bianchini, A.: Pavement maintenance planning at the network level with principal component analysis. J. Infrastruct. Syst. **20**(2), 04013013 (2013)
13. Buddhavarapu, P., Banerjee, A., Prozzi, J.A.: Influence of pavement condition on horizontal curve safety. J. Accid. Analy. Prevent. **52**, 9–18 (2013)
14. Terzi, S.: Modeling for pavement roughness using the ANFIS approach. J. Adv. Eng. Softw. **57**, 59–64 (2013)
15. Alabama Department of Transportation: A Study of Manual Versus Automated Pavement Condition Surveys. Montgomery (2004)
16. Kuhn, K.D.: Pavement network maintenance optimization considering multidimensional condition data. J. Infrastruct. Syst. **18**(4), 270–277 (2012)
17. Alsugair, A.M., Al-Qudrah, A.A.: Artificial neural network approach for pavement maintenance. J. Comput. Civ. Eng. **12**(4), 249–255 (1998)
18. McPherson, E.G., Muchnick, J.: Effects of street tree shade on asphalt concrete pavement performance. J. Arboric. **31**(6), 303–310 (2005)
19. Galleo, E., Moya, M., Piniés, M., Ayuga, F.: Valuation of low volume roads in Spain. Part 2: Methodology validation. J. Biosys. Eng. **101**(1), 135–142 (2008)
20. Li, X., Keegan, K., Yazdani, A.: Index of Foreign Object Damage in Airfield Pavement Management, Transportation Research Record: Journal of the Transportation Research Board, 2153, Transportation Research Board of the National Academies, Washington, D.C., pp. 81–87 (2010)
21. Moazami, D., Behbahani, H., Muniandy, R.: Pavement rehabilitation and maintenance prioritization of urban roads using fuzzy logic. J. Expert Syst. Appl. **38**(10), 12869–12879 (2011)
22. Tabatabaei, S.A., Khaledi, S., Jahantabi, A.: Modeling the deduct value of the pavement condition of asphalt pavement by adaptive neuro fuzzy inference system. J. Pave. Res. Tech. **6** (1), 59–65 (2013)

23. Department of Environment, Heritage and Local Government. (2004). Non-national Roads Condition Study. Ireland
24. Michigan Department of Transportation: Statewide Pavement Management Report: Phase One and Two Airports. Applied Pavement Technology Inc., Illinois (2008)
25. Texas Transportation Institute: Use of Micro Unmanned Aerial Vehicles for Roadside Condition Assessment. Texas Transportation Institute, Texas (2010)
26. Oregon Department of Aviation: Pavement Evaluation/Maintenance Management Program: Final Report—Individual Airports Functional Category 2. Pavement Consultants Inc., Washington (2011)
27. ASTM D6433-11: Standard Practice for Roads and Parking Lots Pavement Condition Index. ASTM international, West Conshohocken (2011)
28. De la Garza, J.M., Piñero, J.C., Ozbek, M.E.: Sampling procedure for performance-based road maintenance evaluations. Transp. Res. Rec.: J. Transp. Res. Board **2044**, 11–18 (2008)
29. Mishalani, R.G., Gong, L.: Optimal infrastructure condition sampling over space and time for maintenance decision-making under uncertainty. J. Transp. Res. Part B **43**(3), 311–324 (2009)
30. Talbi, El: Metaheuristics: from Design to Implementation, 1st edn. Wiley, New Jersey (2009)
31. Miradi, M., Molenaar, A.A., van de Ven, F.C.: Knowledge discovery and data mining using artificial intelligence to unravel porous asphalt concrete in the Netherlands. In: Gopalakrishnan, K., et al. (eds.) Intelligent and Soft Computing in Infrastructure Systems Engineering, SCI, vol. 259, pp. 107–176 (2009)
32. Adeli, H.: Neural networks in civil engineering: 1989-2000. Comput.-Aided Civil Infrastruct. Eng. **16**, 126–142 (2001)
33. Tack, J.N., Chou, E.Y.J.: Multiyear pavement repair scheduling optimization by preconstrained genetic algorithm. Transp. Res. Rec.: J. Transp. Res. Board **1816**, 3–8 (2002)
34. Chan, W.T., Fwa, T.F., Tan, J.Y.: Benefits of information integration in budget planning for pavement management. Transp. Res. Rec.: J. Transp. Res. Board **1889**, 3–12 (2004)
35. Maji, A., Jha, M.K.: Modeling highway infrastructure maintenance schedule with budget constraint, TRB Annual Metting, Transportation Research Board of the National Academies, Washington, D.C (2007)
36. Tang, Y., Chien, S.I.: Scheduling Work Zones for Highway Maintenance Projects Considering a Discrete Time-Cost Relation, Transportation Research Record: Journal of the Transportation Research Board, 2055, pp. 21–30. Transportation Research Board of the National Academies, Washington, D.C. (2008)
37. Gosse, C.A., Smith, B.L., Clarens, A.F.: Environmentally preferable pavement management systems. J. Infrastruct. Syst. **19**(3), 315–325 (2013)
38. Kepaptsoglou, K., Konstantinidou, M., Karlaftis, M.G., Stathopoulos, A.: Planning post-disaster operations in a highway network: a 1 network design model with interdependencies, TRB Annual Metting, Transportation Research Board of the National Academies, Washington, D.C (2014)
39. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
40. Lotfi, M.M., Tavakkoli-Moghaddam, R.: A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems. J. Appl. Soft Comput. **13**(5), 2711–2726 (2013)
41. Cavuoti, S., Garofalo, M., Brescia, M., Paolillo, M., Pescape', A., Longo, G., Ventre, G.: Astrophysical data mining with GPU. A case study: genetic classification. J. New Astro. **26**, 12–22 (2014)
42. Lu, P., Tolliver, D.: Multiobjective pavement-preservation decision making with simulated constraint boundary programming. J. Transp. Eng. **139**(9), 880–888 (2013)
43. Gopalakrishnan, K.: Neural network-swarm intelligence hybrid nonlinear optimization algorithm for pavement moduli back-calculation. J. Transp. Eng. **136**(6), 528–536 (2010)
44. Chang, J.: Particle swarm optimization method for optimal prioritization of pavement sections for maintenance and rehabilitation activities. Appl. Mech. and Mat. **343**, 43–49 (2013)

45. Tayebi, N.R., Moghadas Nejad, F., Mola, M.: Comparison between GA and PSO in analyzing pavement management activities. J. Transp. Eng. **140**(1), 99–104 (2014)
46. Blum, C., Roli, A.: Hybrid Metaheuristics: An Introduction, Studies in Computational Intelligence(SCI), vol. 114, pp. 1–30. Springer, New York (2008)
47. Greene, J., Shahin, M.Y., Alexander, D.R.: Airfield Pavement Condition Assessment, Transportation Research Record: Journal of the Transportation Research Board, No. 1889, TRB, National Research Council, Washington, D.C., pp. 63–70 (2004)
48. Moghadas Nejad, F., Zakeri, H.: The hybrid method and its application to smart pavement management. In: Yang, X. S., Alavi, A. H., (eds.) Metaheuristics in Water, Geotechnical and Transport Engineering, pp. 439–484 (2013)
49. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, Piscataway, N.J. (1995)
50. Kuoa, R.J., Lin., L.M.: Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering. J. Dec. Supp. Sys. **49**, 451–462 (2010)
51. Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 1951–1957 (1999)
52. Parsopoulos, K.E., Vrahatis, M.N.: Recent approaches to global optimization problems through particle swarm optimization. Nat. Comput. **1**, 235–306 (2002)
53. Robinson, J., Rahmat-Samii, Y.: Particle swarm optimization in electromagnetics. IEEE Trans. Antennas Propag. **52**(2), 397–407 (2004)
54. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: Proceedings of IEEE International Conference Systems, Man and Cybernetic. IEEE Service Center, Piscataway, NJ, pp. 4104–4108 (1997)
55. Lipowski, A., Lipowska, D.: Roulette-wheel selection via stochastic acceptance. J. Phys. A **391**(6), 2193–2196 (2012)
56. Lu, Ch., Yu, V.F.: Data envelopment analysis for evaluating the efficiency of genetic algorithms on solving the vehicle routing problem with soft time windows. J. Comput. Indust. Eng. **63**(2), 520–529 (2012)
57. Sivanandam, S.N., Deepa, S.N.: Introduction to Genetic Algorithms, 1st edn. Springer, Berlin (2008)
58. Kao, Y.T., Zahara, E.: A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl. Soft Comput. **8**(2), 849–857 (2008)
59. Zeng, K., Peng, G., Cai, Z., Huang, Z., Yang, X.: A hybrid natural computing approach for the VRP problem based on PSO, GA and quantum computation. In: Yeo, S.-S., et al. (eds.) Computer Science and Its Applications. Springer Science+Business Media, Dordrecht (2012)
60. Castillo, O.: Type-2 Fuzzy Logic in Intelligent Control Applications: A Hybrid PSO-GA Optimization Method to Design Type-1 and Type-2 Fuzzy Logic Controllers, STUDFUZZ 272, pp. 105–118. Springer, Heidelberg (2012)
61. Kumar, P.G., Victoire, T.A., Renukadevi, P., Devaraj, D.: Design of fuzzy expert system for microarray data classification using a novel Genetic Swarm Algorithm. Expert Syst. Appl. **39** (2012), 1811–1821 (2012)
62. Babu, K., Vijayalakshmi, D.: Self-Adaptive PSO-GA hybrid model for combinatorial water distribution network design. J. Pipeline Syst. Eng. Pract. **4**(1), 57–67 (2013)
63. Martínez-Soto, R., Castillo, O., Aguilar, L.T., Rodriguez, A.: A hybrid optimization method with PSO and GA to automatically design Type-1 and Type-2 fuzzy logic controllers, Int. J. Mach. Learn. Cybern. (2013)
64. Galvez, A., Iglesias, A.: A new iterative mutually coupled hybrid GA–PSO approach for curve fitting in manufacturing. Appl. Soft Comput. **13**, 1491–1504 (2013)

Optimum Reinforced Concrete Design by Harmony Search Algorithm

Gebrail Bekdaş, Sinan Melih Nigdeli and Xin-She Yang

Abstract The music-inspired metaheuristic method, called harmony search (HS), is an effective tool in optimization of engineering design problems. HS has been applied for the optimum design of reinforced concrete (RC) members so as to find the best solution, balancing the usability of the design and economy. In this chapter, the optimum design of RC members is presented after optimization of RC members. Then, HS-based optimization applications, such as RC slender columns, RC shear walls, and post-tensioned RC axially symmetric cylindrical walls, are also discussed. The HS-based methods are feasible in finding the optimum design in such problems.

Keywords Metaheuristic methods · Reinforced concrete · Optimization · Harmony search algorithm

1 Introduction

A structural system excited by external forces is under the effect of tensile and compressive stresses. By using brittle materials, like stone bricks, mud brick kiln, and concrete, it will not be possible to construct a structure resisting tensile stresses. These types of materials are effective with their compressive strength, while their

G. Bekdaş · S.M. Nigdeli
Department of Civil Engineering, Istanbul University,
34320 Avcılar, Istanbul, Turkey
e-mail: bekdas@istanbul.edu.tr

S.M. Nigdeli
e-mail: melihnig@istanbul.edu.tr

X.-S. Yang (✉)
Design Engineering and Mathematics, Middlesex University London,
The Burroughs, London NW44BT, UK
e-mail: x.yang@mdx.ac.uk; xy227@cam.ac.uk

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*,
Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_8

tensile strength of the materials is nearly zero. In that case, ductile materials like steel may be used in construction, but the major disadvantages of steel are its high costs and environmental factors. This situation leads us to use composite structures in which two materials with different mechanical behavior are used together. For that reason, optimization is essential in design. Reinforced concrete (RC) structures are the structures using concrete as a form of the whole structure and that form is reinforced with steel bars where tensile stresses occur. In these structures, steel is only used as slender bars while its stability and protection from environmental factor are supported by the concrete covering the bars. The cost minimization problem of RC members has been investigated in several studies. The most challenging issue in RC design is to provide the ductile fracture of concrete. When the fracture is brittle, it will be not possible to absorb huge energy. Thus, the design procedure of RC elements is nonlinear and it is not possible to estimate the best design with the minimum cost. Also, steel bars with fixed sizes in the market and orientation must be suitable for force transfer by forming a bond between steel and concrete. Because of these reasons, numerical iteration methods are more suitable than mathematical ones in the optimum design of RC members.

In finding the optimum design variables of RC members, metaheuristic methods are very suitable. Genetic algorithm (GA) has been employed in several approaches. Coello et al. [1] developed a GA-based approach for RC beams. GA is also used in the methodology of Rafiq and Soutcombe [2] in which biaxial RC columns are optimized. Another study employing GA is the RC design study of Koumousis and Arsenis [3], investigating different types of structural members. Rajaev and Khrisnamoorthy optimized frame structures consisting of RC beam and column members by using GA-based method [4]. Rath et al. [5] used GA in cost optimization while sequential quadratic programing technique is employed in the shape optimization of RC members. Camp et al. investigated the slenderness effect of RC column in the optimization study proposed by for RC frames employing GA [6]. T-shaped beams, investigated together with the effective slab length on the top of the beams are optimized by Ferreira et al. [7] and different design codes are checked.

GA is one of the oldest metaheuristic algorithms and studies combined GA with other methods in their search for the best optimum results. It has been combined with simulated annealing (SA) in order to optimize continuous beams by Leps and Sejnoha [8]. By using a design database, GA was employed in the design of RC frames by Lee and Ahn [9].

Balling and Yao [10] developed a methodology for three dimensional RC frames subjected to different types of loads, such as dead, live, snow, and earthquake. Ahmadkhanlou and Adeli [11] investigated RC slabs by using a methodology with two stages. In the first stage of methodology, the neural dynamics model [12, 13] developed by Adeli and Park is used in order to find the optimum solution of continuous variables and then, perturbation technique is used on modification of the optimum values to practical ones. Several optimization expressions related with the

bending moment, steel area, and ratio of singly or doubly RC beams were developed by Barros et al. [14]. Sirca Jr. and Adeli investigated the minimum cost optimization of prestressed concrete bridges [15]. Govindaraj and Ramasamy [16] optimized continuous beams by selecting design variables from a typical database of reinforcement template. Sahab et al. hybridized GA with the discretized form of Hook and Jeeves method for optimization of RC flat slab building [17]. Statically loaded RC frames were optimized by using GA-based method by Govindaraj and Ramasamy [18]. Optimization of multi-bay single-story and single-bay multi-story RC frames are done by Guerra and Kiousis [19].

Simulated annealing (SA) is also a widely employed algorithm in RC design optimization. Paya et al. [20] employed SA in RC frame optimization. SA is also used as a part in the methodology of Perea et al. [21]. In this research, RC frames of bridges were optimized by using SA with a metaheuristic called threshold accepting and two heuristic methods such as random walk and descent local search. SA is also employed in the RC frame optimization study considering the minimization of the value of embedded CO_2 emission by Paya-Zaforteza et al. [22]. Also, Camp and Huq employed big bang big crunch algorithm (BB-BC) for the CO_2 emission minimization of RC frames [23].

A reinforcement sizing diagram was provided for RC beams and columns by Gil-Martin et al. [24]. The optimum depth and reinforcement design of rectangular RC beams was investigated by Barros et al. [25]. Fedghouche and Tiliouine optimized T-shaped RC beam by employing GA and the study only covers the design of beams without steel bars under compressive forces [26].

The optimum design of RC retaining walls is also challenging because stability of the wall subjected to soil load must be ensured. This security is generally provided by increasing the weight of the wall, but in that case, the internal forces increase and the economy cannot be provided. For that reason, a balanced optimum design must be found and metaheuristic methods, such as SA [27, 28], harmony search (HS) [29], BB-BC [30], and charged system search (CSS) [31] have been successfully employed for optimization of RC retaining walls.

HS algorithm has also been employed for optimum RC members, such as continuous beams [32], T-shaped beams [33], columns [34, 35], frames [36], shear walls [37], cylindrical walls [38], post tensioning cylindrical walls [39], and biaxial columns [40]. Kaveh and Sabzi used the combination of several metaheuristic in the optimum design of RC frames [41]. BB-BC-based RC beam optimization was developed by Kaveh and Sabzi [42]. SA and Tabu search were combined in order to develop an optimum design methodology for hybrid fiber reinforced composite plates by Rama Mohan Rao and Shyju [43]. Nigdeli and Bekdaş developed a random search technique in optimization of RC beams [44]. Bekdaş and Nigdeli optimized uniaxial RC beam by using random search technique [45, 46]. Jahjouh et al. investigated continuous beams by employing Artificial Bee Colony (ABC) [47].

2 Harmony Search Algorithm

As designers searching for the optimum design, a musician also has a primary goal, which is to achieve the harmony of the music and thus, the admiration of the audience. The Harmony Search (HS) algorithm is a music-inspired algorithm, developed by Geem et al. [48]. HS is a memory-based random search method. In the musical performances, musicians try to find a pleasing harmony and this harmony is a perfect state of music performance. Similarly, researchers try to find a global solution (a good, ideally, optimal solution of all design variables) for minimizing or maximizing an objective function (or objective functions in multi-objective optimization). HS uses a stochastic random search instead of a gradient search. HS was extended to solve both discrete variable [49] and continuous variable [50] problems.

The HS method can be briefly explained in five steps for design problems.

i. In the first step, constant values of the optimization problem are defined. These values are solution ranges of design variables, specific parameters of HS algorithm, and design problem constants. The solution ranges are defined in order to find the solution quickly by tightening the range or preventing several solution domains for design constraints or practical design. The algorithm parameters are harmony memory size (HMS), harmony memory considering rate (HMCR), and pitch adjusting rate (PAR). Additionally, the HS algorithm was also improved in order to develop parameter setting free techniques [51, 52].
ii. Before using the properties of the algorithm, an initial harmony memory (HM) matrix is constructed and it contains harmony vectors. The parameter; HMS is the number of harmony vectors in HM and a harmony vector includes a set of design variables, which are randomly generated by using the solution ranges defined in the first step. Also, the optimization objective or objectives are defined for each set of design variables. The formulation must contain the consideration of design constraints if the constraints are violated. In design problems, design constraints are generally seen and the exiting of these constraints may be the reason of using numerical optimization methods.
iii. In this step, the optimization starts. A new harmony vector is defined by using the special rules of HS. In musical performances, musicians play a random notes and this note may be a current popular song. In that case, a musician may slightly chance this note in order to gain admiration by playing notes similar to current best. Or maybe, a musician may try to play a new melody in a performance. Musicians must use these two options. Otherwise, they may lose their popularity if they do not produce a novel work. In optimum design, we need to search the best result around the current best solution in order to improve the convergence of the method, but the solution should not be a local optimum, though there is no guarantee. For that reason, we need to generate new solutions by using randomization in the whole domain. A new vector is generated around the existing ones in HM or from the initial range. The value of HMCR controls the type of the generation and it is the possibility of the

generation done by using an existing vector in HM. The neighboring values of a chosen vector may be defined by using a smaller solution range. The parameter; PAR is used to reduce the solution area of design variables. The ratio of the small range and initial range can be defined with the value of PAR. The limits of the design variables should also be applied to ensure the solution ranges. If the chosen vector is near to the bounds of solution ranges, the design variables may be assigned with the values out of the range.

iv. In this step, the newly generated harmony vector is selected or not. It is compared with the current worst vector in mean of minimization or maximization of objective functions. If the new one is better than the worst one, the new solution is stored by eliminating the worst one. Thus, the harmony memory is updated.

v. In the last step, the termination criterion or criteria are changed. Until this termination criterion is provided, the optimization process continues from the third step. The termination criterion may be a fixed number of iterations or convergence of the design variables. It is possible to use several termination criteria in design problems.

3 Optimum Design Examples

In this section, three applications of RC design members are presented by employing the HS algorithm. The problems are the optimization of slender RC columns, optimization of RC shear walls, and optimum design of post-tensioned RC axially symmetric cylindrical walls.

3.1 Optimum Design of Slender RC Columns

In a structural system, columns are generally under the effect of axial forces. Due to the deflection of columns under external loads, second-order effects occur, and slenderness and buckling of the columns is the reason of this issue. Slenderness and buckling stress can be affected by the length of the column, cross-sectional dimensions and effective length factor in buckling (k) of columns. In this example, HS is employed by considering the regulations of ACI318 (Building Code Requirements for Structural Concrete) [54].

In ACI318, slenderness effects are considered by using the approximate design procedure in which the flexure moment on the column is magnified with moment magnification factor (δ_s). In this procedure, the moments of inertia of beams and columns are, respectively, reduced with 65 and 30 % because of consideration of cracking. Subject to the ductile design, cracking of the beam is more possible than columns and, for that reason, the reduction of the moments of inertia of beams can

be big. Effective length factors in buckling (k) are found according ratio; Ψ calculated at both ends of columns (Ψ_A; upper end—Ψ_B; lower end). In the equation of this ratio, E, I, and l are elasticity modulus, moment of inertia, and the length of the corresponding RC members, respectively, as seen in Eq. (1).

$$\Psi_{A,B} = \frac{\sum (\mathrm{EI}/l)_{\mathrm{column}}}{\sum (\mathrm{EI}/l)_{\mathrm{beam}}} \tag{1}$$

After Ψ_A and Ψ_B are calculated, k is obtained from the set of equations given as Eq. (2). These equations are for structural systems which are free in horizontal direction.

$$\left.\begin{aligned} \Psi_m &= 0.5(\Psi_A + \Psi_B) \\ k &= \frac{20 - \Psi_m}{20}\sqrt{1+\Psi_m} \quad if\ \Psi_m < 2 \\ k &= 0.9\sqrt{1+\Psi_m} \quad\quad\ if\ \Psi_m \geq 2 \end{aligned}\right\} \tag{2}$$

The moment magnification factor (δ_s) is calculated as

$$\delta_s = \frac{C_m}{1 - \frac{P_u}{0.75P_c}} \tag{3}$$

where C_m is the correction factor which considers the actual moment diagram to an equivalent uniform moment diagram. It can be calculated as

$$C_m = 0.6 + 0.4\frac{M_1}{M_2} \tag{4}$$

and C_m must be bigger than 0.4, while it is 1.0 for members with transverse loads between supports. In Eq. (3), critical buckling load is calculated as

$$P_c = \frac{\pi^2 EI}{(kl)^2} \tag{5}$$

by reducing the rigidity of the column by 75 %.

The design variables of the problem are the dimensions of the column and amount of reinforcements (both longitudinal and shear). The column is under loadings, such as flexural moment (M), shear force (V), and axial force (N). The design constants are clear cover of concrete (c_c), maximum aggregate diameter (D_{max}), length of column (l), elasticity modulus of steel (E_s), cost of the concrete per m^3 (C_c), cost of the steel per ton (C_s), compressive strength of concrete $\left(f_c'\right)$, yield strength of steel (f_y), specific gravity of steel (γ_s), and specific gravity of concrete (γ_c) are defined.

After the cross-section dimensions are randomized, ductility conditions given in Eqs. (6) and (7) are checked. A_c is the area of the cross section of columns.

$$V < \begin{cases} 0.2f'_c A_c \\ 5.5A_c \end{cases} \tag{6}$$

$$N < 0.5f'_c A_c \tag{7}$$

If the element is under brittle fracture risks, the cross-sectional dimensions are updated before the randomization of reinforcements are done. Thus, the design constraints, given as Eqs. (6) and (7), are considered and the computation time is saved. Then, the number and size of reinforcement bars are done and placement conditions are checked. If the placement requirements given in ACI318;

$$a_\phi > \begin{cases} 1.5\phi_{avarage} \\ 40 \text{ mm} \\ \frac{4}{3} D_{max} \end{cases} \tag{8}$$

is not provided, reinforcements are updated by also considering a design with two lines. $\phi_{average}$ is the average of the diameter sizes and a_ϕ is the clear distance between reinforcement bars. Similarly, as done in ductility conditions, the reinforcements are iteratively randomized if the placement conditions are not provided.

In order to conduct a design for axial forces and magnified flexural moments, a design methodology using random searches is conducted in production of harmony vectors. The distance from extreme compressive fiber to neural axis (c) is iteratively scanned for the best flexural moment and axial force combination. For the required axial force, the corresponding flexural moment capacity is found. This flexural moment capacity must be close and higher than the required one. The difference of the percentage may be defined as a function of the iteration number. Thus, the solutions far from the optimum are quickly eliminated.

After the design for axial force and magnified flexural moment, the design of shear reinforcements is done by iteratively assigning diameter sizes. The required distance of shear reinforcement (stirrups) is calculated for all iterations. Then, the nominal shear strength of concrete (V_c) and nominal shear strength of reinforcement (V_s) are calculated as given in Eqs. (9) and (10), respectively.

$$V_c = \frac{\sqrt{f'_c}}{6} b_w d \tag{9}$$

$$V_s = \frac{A_v f_y d}{s} \tag{10}$$

where A_v represents the area of shear reinforcement spacing s. The effective depth of the column is defined with d. Additionally, the V_s value must be lower than $0.66\sqrt{f_c'}b_w d$. If this constraint and total shear strength of the column is lower than the required one, the objective function is penalized. Also, the calculated results must be higher than the minimum shear reinforcement ($A_{v,min}$) value and the distance between the bars must be lower than the maximum shear reinforcement distance (s_{max}) given in Eqs. (11) and (12), respectively

$$(A_v)_{min} = \frac{1}{3}\frac{b_w s}{f_y} \tag{11}$$

$$s_{max}\begin{cases} \leq \dfrac{d}{4} & \text{if } V_s \geq 0.33\sqrt{f_c'}b_w d \\ \leq \dfrac{d}{2} & \text{otherwise.} \end{cases} \tag{12}$$

The total material cost of design (C), which is the main objective of the problem, is also stored in the harmony vectors. The elimination of the worst solutions is done according to the same objective function given in Eq. (13). The optimization process is done for the maximum iteration number. A_s, u_{st}, γ_s, C_c, and C_s are the area of non-prestressed longitudinal reinforcement, the length of shear reinforcement spacing s, the specific gravity of steel, the material cost of the concrete per m^3 and the material cost of the steel per ton, respectively.

$$\min C = (A_c - A_s)C_c + (A_s + \frac{A_v}{s}u_{st})l\gamma_s C_s \tag{13}$$

The loading at the upper end of the column such as axial force, flexural moment, and shear force were taken as 2000 kN, 50 kNm, and 50 kN, respectively. Design constants are taken as given in Table 1.

Optimum design variables are investigated by using k values between 0.5 and 2.0. The optimum results are presented in Table 2. The slenderness effects are not effective for columns with an effective length factor in buckling (k) lower than 1.3. In that situation, the optimum values are equal for all. Then, parallel to k, the increase of the cost is seen.

3.2 Optimum Design of RC Shear Walls

In this example, RC shear walls are presented for optimum design considering the total cost minimization and the design methodology is done according to ACI 318 [53]. The optimum design is scanned by randomizing the design variable show in Fig. 1. The design variables are breadth of the shear wall (b_w), reinforcements in

Table 1 The design constants of column example

Definition	Symbol	Unit	Value
Range of the breadth	b_w	mm	250–400
Range of the height	h	mm	300–600
Length	l	mm	4500
Clear cover	c_c	mm	30
Range of reinforcement	ϕ	mm	16–30
Range of shear reinforcement	ϕ_v	mm	8–14
Max. aggregate diameter	D_{max}	mm	16
Yield strength of steel	f_y	MPa	420
Comp. strength of concrete	f'_c	MPa	25
Elasticity modulus of steel	E_s	MPa	200000
Specific gravity of steel	γ_s	t/m^3	7.86
Cost of the concrete per m^3	C_c	$	40
Cost of the steel per ton	C_s	$	400

Table 2 Optimum results for column example

k	0.5–1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
Breadth of column (b_w) (mm)	300	300	300	300	300	300	350	350	350
Height of column (h) (mm)	550	550	600	600	600	600	600	600	600
Bars in each face	1Φ18+ 1Φ16	3Φ16	2Φ16	2Φ16	2Φ16	2Φ16+ 1Φ18	1Φ20+ 1Φ18	1Φ20+ 1Φ18	4Φ16
Web reinforcements	2Φ16	2Φ16	2Φ18	2Φ18	2Φ18	2Φ16	2Φ18	2Φ18	2Φ16
Shear reinforcement diameter (mm)	Φ8	Φ8	Φ8	Φ8	Φ8	Φ8	Φ8	Φ8	Φ8
Shear reinforcement distance (mm)	240	240	270	270	270	270	270	270	270
Optimum cost ($)	58.764	62.889	62.972	62.972	62.972	67.097	73.308	73.308	76.906

columns headings and web of the shear walls. All reinforcements including the stirrups are defined with a diameter size in mm and a distance between each bars in mm. The proposed methodology can be by the following steps.

First step: As similarly for all design optimization problems, design constants shown in Table 3, the ranges of design variables (also shown in Table 3), special harmony search parameters explained in the second section and the design internal forces of the RC shear walls (as given in Table 4) are defined.

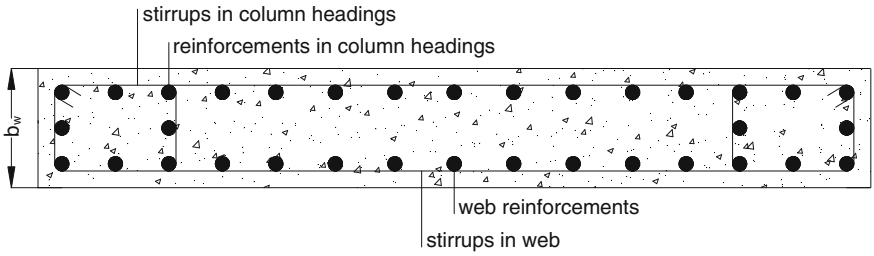

Fig. 1 Design variables of RC shear wall

Table 3 The design constants and solution ranges of RC shear wall example

Definition	Symbol	Unit	Value in example
Range of the breadth	b_w	mm	250–600
Height	h	mm	3000
Length	l	mm	4000
Clear cover	c_c	mm	30
Range of reinforcement	ϕ	mm	16–30
Range of shear reinforcement	ϕ_v	mm	8–14
Max. aggregate diameter	D_{max}	mm	16
Yield strength of steel	f_y	MPa	420
Comp. strength of concrete	f'_c	MPa	25
Elasticity modulus of steel	E_s	MPa	200000
Specific gravity of steel	γ_s	t/m^3	7.86
Specific gravity of concrete	γ_c	t/m^3	2.5
Cost of the concrete per m^3	C_c	\$/m^3	40
Cost of the steel per ton	C_s	\$/t	400

Second step: The generations of harmony vectors are done in three substeps named as steps a, b, and c. In these steps, iterative randomization is done in order to ensure the ACI 318 requirements.

Step a: The breadth (b_w) of the shear wall is randomized and ductility conditions given in Eqs. (6–7) are checked. The randomization of b_w continues until the ductility conditions are met. Also, some discrete randomization is carried out in order to assign values which are the multiples of 50 mm. In the construction yard, a sensible production of cross-section dimensions is not possible.

Step b: Then, the reinforcements are randomized by checking the capacity of the shear wall. The reinforcements at the column headings and web are separately randomized. For the design shear force (V), shear reinforcement (stirrup) design is also carried out by the procedure given in the RC column example.

Step c: In the last substep, the distance from extreme compressive fiber to neural axis (c) is iteratively searched in order to find the best flexural moment and axial force combination. The stress on each bar is calculated and the force equilibrium is

Table 4 Internal forces for three cases of RC shear wall

	Axial force [N (kN)]	Flexural moment [M (kNm)]	Shear force [V (kN)]
Case 1	500	25	100
Case 2	1000	50	200
Case 3	2000	100	400

provided in the calculation of the capacity of the shear wall. The stress on concrete section is calculated by assuming an equaling rectangular stress block as explained in ACI 318. The same procedure with the first example is carried out in order to find the best suitable design ensuring the axial force and the flexural moment capacity. Finally, the total material cost of the shear wall is calculated and the cost is penalized, if the flexural capacity of the design is lower than the required one after several randomizations. In that case, the breadth of the design may be too small for the design forces.

By using the three substeps including random searches, the optimization is decreased by eliminating the results far from the optimum one. Thus, it is not needed to use previously generated templates for reinforcement design. Generally, the final results in harmony vectors are good designs which are physically possible solutions and close to optimum design. By using the rules of HS, the best one is searched in the third step.

Third step: In this step, the initial harmony memory matrix is updated by using the rules of HS. The three substeps are also used in the generation of new harmony vectors.

Last step: The stopping criterion is the maximum number of iterations. When the number of iterations reaches to the maximum iteration number defined by the user, the optimization process is terminated. The maximum iteration number is 2000 for three loading cases, as given in Table 4.

The optimum design variables are given in Table 5. From the results, the importance of optimization is clearly seen that the optimum design variables are far away from the lower and upper bounds of the solution ranges. Concrete is a cheaper material than steel, but the optimum reinforcements are not at the minimum limit. For that reason, the required optimum with a balance between both materials is found and the RC shear wall design with metaheuristic methods is a feasible approach.

Table 5 Optimum design variables of RC shear wall

Cases	b_w (mm)	Reinforcements at the column heading	Web reinforcements of column heading	Web reinforcement of shear wall	Stirrups	Cost ($)
1	250	1Φ20 + 1Φ16	1Φ18 + 1Φ16	3Φ18 + 1Φ16	Φ10/160	232.2871
2	300	1Φ26 + 1Φ22	3Φ16	1Φ18 + 3Φ16	Φ10/120	295.9118
3	450	1Φ18 + 1Φ16	1Φ18 + 1Φ16	2Φ18 + 2Φ16	Φ8/110	324.2167

3.3 Optimum Design of Cylindrical Walls

For the third example, cost optimization of post-tensioned RC axially symmetric cylindrical walls are presented. A typical cylindrical wall model can be seen in Fig. 2. The optimization process can be summarized in three stages; static analyses, RC design/cost calculation, and comparison/elimination.

Static analyses: After defining the geometry (height: H and radius of wall: r) of wall, specific weight of the liquid, design constant, such as, material cost of material, workmanship, yield strength of steel, parameters of HS algorithm, and design variables intensities (P_1, P_2…, P_n) and locations (a_1, a_2…, a), thickness of wall (h), compressive strength of concrete (f'_c), the structural analyses are done by using the superposition method [54] to obtain internal forces, flexural moments, shear force, and axial tension.

RC design and cost calculations: In this stage, design constraints; flexural moment capacities in both direction, shear strength capacity, axial tension, minimum steel ratios, minimum and maximum bar spacing, maximum crack width are checked by considering the diameter of horizontal (ϕ_h), and vertical (ϕ_v) bars and bar spacings in both direction (S_h and S_v) as design variables. By providing safety and applicable solution, the total cost of the wall as objective function of process is calculated as

$$\min f(X) = C_c V_c + C_s W_s + C_{pt} W_{pt} + C_{fw} A_{fw} \tag{14}$$

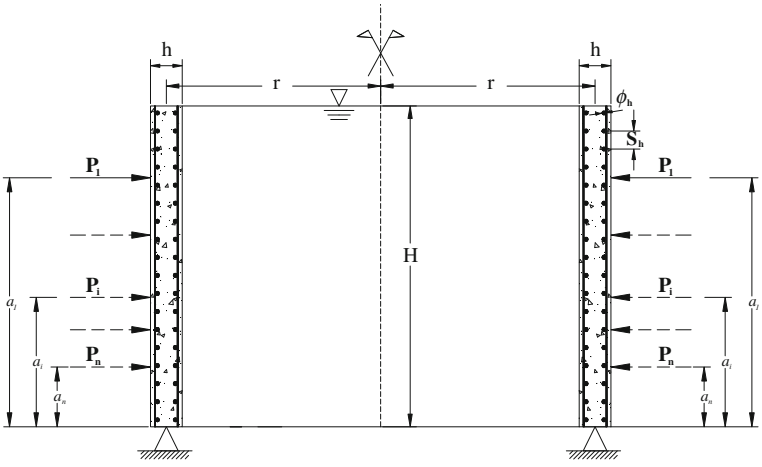

Fig. 2 Model of cylindrical wall used in optimization process

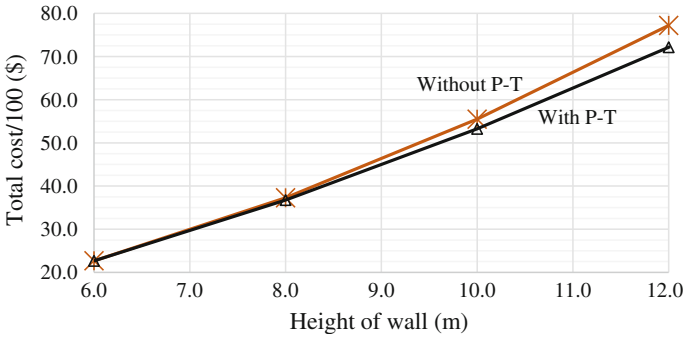

Fig. 3 The optimum results of the cylindrical wall

where

$$V_c = \pi H\left((r+\frac{h}{2})^2 - (r-\frac{h}{2})^2\right) \tag{15}$$

$$W_s = \frac{\pi\phi_v^2}{4}\left(\frac{4\pi r}{S_v}\right)H\gamma_s + \frac{\pi\phi_h^2}{4}\left(\frac{2H}{S_h}\right)2\pi r\gamma_s \tag{16}$$

$$W_{pt} = \frac{\pi\sum_{i=1}^{n}(\phi_{pt})_i^2}{4}2\pi r\gamma_s \tag{17}$$

$$A_{fw} = 2\pi H(2r) \tag{18}$$

Comparison and elimination: In this stage, comparisons and eliminations are done according to rules of HS algorithm.

The optimization process is performed for five different wall models with different heights considering with and without post-tensioning (P-T). Comparing the results given in Fig. 3, although being limited difference between the costs for 6 m and 8 m wall heights (0.09 % and 1.42 %), by increasing wall height as well as loads, the effects of post-tensioning loads can be clearly seen (4.05 % and 6.55 %).

4 Conclusion

In this chapter, we have described the optimum designs of RC members using HS search. The objective function and complex constraints have been formulated for each case study. Numerical experiments have been carried out and results have been analyzed.

From the above examples, we can see that HS can be a good tool to provide designs with quality solutions that can be used in practical applications. Obviously, it would be useful to see how other algorithms perform and make a detailed comparison for these types of problems.

References

1. Coello, C.C., Hernandez, F.S., Farrera, F.A.: Optimal design of reinforced concrete beams using genetic algorithms. Expert Syst. Appl. **12**, 101–108 (1997)
2. Rafiq, M.Y., Southcombe, C.: Genetic algorithms in optimal design and detailing of reinforced concrete biaxial columns supported by a declarative approach for capacity checking. Comput. Struct. **69**, 443–457 (1998)
3. Koumousis, V.K., Arsenis, S.J.: Genetic algorithms in optimal detailed design of reinforced concrete members. Comput-Aided Civ. Inf. **13**, 43–52 (1998)
4. Rajeev, S., Krishnamoorthy, C.S.: Genetic algorithm-based methodology for design optimization of reinforced concrete frames. Comput-Aided Civ. Inf. **13**, 63–74 (1998)
5. Rath, D.P., Ahlawat, A.S., Ramaswamy, A.: Shape optimization of RC flexural members. J. Struct. Eng.-ASCE **125**, 1439–1446 (1999)
6. Camp, C.V., Pezeshk, S., Hansson, H.: Flexural design of reinforced concrete frames using a genetic algorithm. J. Struct. Eng.-ASCE **129**, 105–111 (2003)
7. Ferreira, C.C., Barros, M.H.F.M., Barros, A.F.M.: Optimal design of reinforced concrete T-sections in bending. Eng. Struct. **25**, 951–964 (2003)
8. Leps, M., Sejnoha, M.: New approach to optimization of reinforced concrete beams. Comput. Struct. **81**, 1957–1966 (2003)
9. Lee, C., Ahn, J.: Flexural design of reinforced concrete frames by genetic algorithm. J. Struct. Eng.-ASCE **129**(6), 762–774 (2003)
10. Balling, R., Yao, X.: Optimization of reinforced concrete frames. J. Struct. Eng.-ASCE **123**, 193–202 (1997)
11. Ahmadkhanlou, F., Adeli, H.: Optimum cost design of reinforced concrete slabs using neural dynamics model. Eng. Appl. Artif. Intell. **18**(1), 65–72 (2005)
12. Adeli, H., Park, H.S.: Optimization of space structures by neural dynamics. Neural Netw. **8**(5), 769–781 (1995)
13. Adeli, H., Park, H.S.: Neurocomputing for Design Automation. CRC Press, Boca Raton (1998)
14. Barros, M.H.F.M., Martins, R.A.F., Barros, A.F.M.: Cost optimization of singly and doubly reinforced concrete beams with EC2-2001. Struct. Multidiscip. O. **30**, 236–242 (2005)
15. Sirca Jr, G., Adeli, H.: Cost optimization of prestressed concrete bridges. J. Struct. Eng. ASCE **131**(3), 380–388 (2005)
16. Govindaraj, V., Ramasamy, J.V.: Optimum detailed design of reinforced concrete continuous beams using Genetic Algorithms. Comput. Struct. **84**, 34–48 (2005)
17. Sahab, M.G., Ashour, A.F., Toropov, V.V.: Cost optimisation of reinforced concrete flat slab buildings. Eng. Struct. **27**, 313–322 (2005)
18. Govindaraj, V., Ramasamy, J.V.: Optimum detailed design of reinforced concrete frames using genetic algorithms. Eng. Optimiz. **39**(4), 471–494 (2007)
19. Guerra, A., Kiousis, P.D.: Design optimization of reinforced concrete structures. Comput. Concrete. **3**, 313–334 (2006)
20. Paya, I., Yepes, V., Gonzalez-Vidosa, F., Hospitaler, A.: Multiobjective optimization of concrete frames by simulated annealing. Comput-Aided Civ. Inf. **23**, 596–610 (2008)
21. Perea, C., Alcala, J., Yepes, V., Gonzalez-Vidosa, F., Hospitaler, A.: Design of reinforced concrete bridge frames by heuristic optimization. Adv. Eng. Softw. **39**, 676–688 (2008)

22. Paya-Zaforteza, I., Yepes, V., Hospitaler, A., Gonzalez-Vidosa, F.: CO_2-optimization of reinforced concrete frames by simulated annealing. Eng. Struct. **31**, 1501–1508 (2009)
23. Camp, C.V., Huq, F.: CO_2 and cost optimization of reinforced concrete frames using a big bang-big crunch algorithm. Eng. Struct. **48**, 363–372 (2013)
24. Gil-Martin, L.M., Hernandez-Montes, E., Aschheim, M.: Optimal reinforcement of RC columns for biaxial bending. Mater. Struct. **43**, 1245–1256 (2010)
25. Barros, A.F.M., Barros, M.H.F.M., Ferreira, C.C.: Optimal design of rectangular RC sections for ultimate bending strength. Struct. Multidiscip. O. **45**, 845–860 (2012)
26. Fedghouche, F., Tiliouine, B.: Minimum cost design of reinforced concrete T-beams at ultimate loads using Eurocode2. Eng. Struct. **42**, 43–50 (2012)
27. Ceranic, B., Freyer, C., Baines, R.W.: An application of simulated annealing to the optimum design reinforced concrete retaining structure. Comput. Struct. **79**, 1569–1581 (2001)
28. Yepes, V., Alcala, J., Perea, C., Gonzalez-Vidosa, F.: A parametric study of optimum earth-retaining walls by simulated annealing. Eng. Struct. **30**, 821–830 (2008)
29. Kaveh, A., Abadi, A.S.M.: Harmony search based algorithms for the optimum cost design of reinforced concrete cantilever retaining walls. Int. J. Civil Eng. **9**(1), 1–8 (2011)
30. Camp, C.V., Akin, A.: Design of retaining walls using big bang–big crunch optimization. J. Struct. Eng.-ASCE, **138**(3), 438–448 (2012)
31. Talatahari, S., Sheikholeslami, R., Shadfaran, M., Pourbaba, M.: Optimum design of gravity retaining walls using charged system search algorithm. Math. Probl. Eng. **2012**, 1–10 (2012)
32. Akin, A., Saka, M.P.: Optimum detailed design of reinforced concrete continuous beams using the harmony search algorithm. In: Topping, B.H.V., Adam, J.M., Pallarés, F.J., Bru, R., Romero, M.L. (eds.) Proceedings of the Tenth International Conference on Computational Structures Technology, Civil-Comp Press, Stirlingshire, UK, Paper 131 (2010). doi:10.4203/ccp.93.131
33. Bekdaş, G., Nigdeli, S.M.: Optimization of T-shaped RC flexural members for different compressive strengths of concrete. Int. J. Mech. **7**, 109–119 (2013)
34. Bekdaş, G., Nigdeli S.M.: Optimization of slender reinforced concrete columns. 85th annual meeting of the international association of applied mathematics and mechanics, 10–14 March 2014, Erlangen, Germany (2014)
35. Nigdeli, S.M., Bekdaş, G.: Optimum design of RC columns according to effective length factor in buckling. the twelfth international conference on computational structures technology, 2–5 Sept 2014, Naples, Italy
36. Bekdaş, G., Nigdeli, S.M.: Optimization of RC frame structures subjected to static loading. In: 11th World Congress on Computational Mechanics, 20–25 July 2014, Barcelona, Spain (2014)
37. Nigdeli, S.M., Bekdaş, G.: Optimization of reinforced concrete shear walls using harmony search. In: 11th International Congress on Advances in Civil Engineering, 21–25 Oct 2014, Istanbul, Turkey (2014)
38. Bekdaş, G.: Harmony search algorithm approach for optimum design of post-tensioned axially symmetric cylindrical reinforced concrete walls. J. Optim. Theory Appl. **164**(1), 342–358 (2015)
39. Bekdas, G.: Optimum design of axially symmetric cylindrical reinforced concrete walls. Struct. Eng. Mech. **51**(3), 361–375 (2014)
40. Nigdeli, S.M., Bekdaş, G., Kim, S., Geem, Z.W.: A novel harmony search based optimization of reinforced concrete biaxially loaded columns. Struct. Eng. Mech. doi:http://dx.doi.org/10.12989/sem.2015.54.6.000
41. Kaveh, A., Abadi, A.S.M.: Harmony search based algorithms for the optimum cost design of reinforced concrete cantilever retaining walls. Int. J. Civil Eng. **9**(1), 1–8 (2011)
42. Kaveh, A., Sabzi, O.: Optimal design of reinforced concrete frames using big bang-big crunch algorithm. Int. J. Civil Eng. **10**(3), 189–200 (2012)
43. Rama Mohan Rao, A.R., Shyju, P.P.: A meta-heuristic algorithm for multi-objective optimal design of hybrid laminate composite structures. Comput-Aided Civ. Infrastruct. Eng. **25**(3), 149–170 (2010)

44. Nigdeli, S.M., Bekdas, G.: Optimization of RC beams for various cost ratios of steel/concrete. In: 4th European Conference of Civil Engineering ECCIE'13, 8–10 Oct 2013, Antalya, Turkey (2013)
45. Bekdaş, G., Nigdeli, S.M.: Optimum design of uniaxial RC columns. In: An International Conference on Engineering and Applied Sciences Optimization, 4–6 June 2014, Kos Island, Greece (2014)
46. Bekdaş, G., Nigdeli, S.M.: Optimization of reinforced concrete columns subjected to uniaxial loading. In: Engineering and Applied Sciences Optimization, pp. 399–412. Springer International Publishing (2015)
47. Jahjouh, M.M., Arafa, M.H., Alqedra, M.A.: Artificial Bee Colony (ABC) algorithm in the design optimization of RC continuous beams. Struct Multidiscip. Optim. **47**(6), 963–979 (2013)
48. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. Simulation **76**, 60–68 (2001)
49. Lee, K.S., Geem, Z.W., Lee, S.H., Bae, K.W.: The harmony search heuristic algorithm for discrete structural optimization. Eng. Optim. **37**, 663–684 (2005)
50. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. Comput. Methods Appl. Mech. Eng. **194**, 3902–3933 (2005)
51. Geem, Z.W., Sim, K.-B.: Parameter setting free harmony search algorithm. Appl. Math. Comput. **217**, 3881–3889 (2010)
52. Hasancebi, O., Erdal, F., Saka, M.P.: Adaptive harmony search method for structural optimization. J. Struct. Eng. **136**, 419–431 (2010)
53. ACI 318M-05, *Building code requirements for structural concrete and commentary*, American Concrete Institute, 2005
54. Hetenyi, M.: Beams on Elastic Foundation. The University of Michigan Press, Ann Arbor (1946)

Reactive Power Optimization in Wind Power Plants Using Cuckoo Search Algorithm

K.S. Pandya, J.K. Pandya, S.K. Joshi and H.K. Mewada

Abstract This chapter presents the application of a new meta-heuristic optimization algorithm called cuckoo search algorithm (CSA) to solve optimal reactive power dispatch problem (ORPD) of the power system in the presence of wind power plants (WPP). Due to the inclusion of WPP, the ORPD problem becomes a complex combinatorial optimization problem and it has a nonlinear objective function with many local minima, and discontinuous and nonlinear constraint functions. CSA is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lèvy flight behavior of some birds and fruit flies. The effectiveness and feasibility of CSA have been tested on a 41-bus WPP test system and the obtained results that have been compared with particle swarm optimization (PSO). Simulation results yield that the CSA converges to better optimal solutions faster than PSO.

Keywords Artificial intelligence method · Cuckoo search algorithm · Particle swarm optimization · Reactive power optimization · Wind power plant

K.S. Pandya (✉)
Department of Electrical Engineering, CSPIT,
Charotar University of Science and Technology, Changa, India
e-mail: kartikpandya.ee@charusat.ac.in

J.K. Pandya
Department of Civil Engineering, Dharmsinh Desai University, Nadiad, India
e-mail: jignapandya2011@gmail.com

S.K. Joshi
Department of Electrical Engineering,
The M.S. University of Baroda, Vadodara, India
e-mail: skjoshi@ieee.org

H.K. Mewada
Department of Electronics & Communications, CSPIT,
Charotar University of Science and Technoloogy, Changa, India
e-mail: hirenmewada.ec@charusat.ac.in

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*,
Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_9

1 Introduction and Literature Survey

The penetration of wind power plants into conventional power systems has been increased drastically all over the world in recent years to ensure higher reliability, security, stability, and to fulfill ever increasing load demands of the power systems. However, the electricity grid codes [1] of many utilities suggest that the wind farms should be able to supply a sufficient amount of active power, reactive power to the power grid, and should maintain a power factor between 0.95 lagging to 0.95 leading at the point of common coupling (PCC). To maintain the power factor within the required feasible range, it is necessary to execute reactive power management by the transmission system operator (TSO). So in this context, TSO performs the optimal reactive power dispatch (ORPD) in which it tries to "optimally" set the values of the control variables including the reactive power output of the generators (generator bus voltages), tap ratios of transformers, and reactive power output of shunt compensators like capacitors/inductors so as to minimize the total transmission active power losses, while satisfying a given set of constraints.

Up to now, many classical optimization methods and artificial intelligence (AI) methods have been used to solve ORPD problems of small- and large-scale power systems. Deeb et al. [2] proposed linear programming (LP) with a decomposition approach to solve ORPD problem. Even though LP method obtains results in less time, it does not yield a global solution in case of a large power system. It is because of nonlinear objective function and constraints have to be linearized to overcome the inherent limitations of LP method. Gomez et al. [3] introduced the concept of "Decomposition" in which continuous problem and integer problems are solved separately. As a result, the computational time required is decreased. Wu et al. [4] suggested nonlinear predictor-corrector primal-dual interior point method to solve ORPD problem. Kermanshahi et al. [5] used successive quadratic programming (SQP) to optimize the allocation of reactive power sources. However, this method requires inversion of second-order partial derivative matrix (Hessian matrix) in each iteration, which is quite time consuming. Chattopadhyay et al. [6] included security margin(SM) as the constraint in optimal reactive power planning to ensure that the operating point should remain at least at a "SM" distance in post-contingency case. However, this approach will become time consuming in medium or large power systems as a large number of VAR support configurations have to be considered. Yan et al. [7] proposed predictor-corrector primal-dual interior point algorithm to solve ORPD problem. The Hessian matrix in this method remains constant and needs to be evaluated only once in the entire optimization process. Rabiee et al. [8] introduced the concept of "local voltage stability index" to incorporate it into objective function of loss minimization. The objective function based upon this index yields some improvement in voltage stability margins.

When classical methods are applied to solve small-scale power systems (i.e., with less than 30 buses), they can converge to the optimal solution in a reasonable time. Also, all the methods give almost same optimal solutions. However, when

applied to the medium- or large-scale power systems (i.e., with more than 30 buses), they suffer with the following drawbacks:

1. They may not find the global optimal solution and may get trapped into local minima.
2. The choice of initial starting point (Initial guess) greatly affects the convergence and as a result different optimal solutions are obtained.
3. Due to zigzagging in the search direction, the speed of these methods decreases and thus a large number of iterations are required to get the optimal solution.
4. Linearization of the constraints and objective functions may reduce the accuracy of the answer.
5. Hessian matrix may become singular for large-scale power system.

Due to these mentioned reasons, the applications of the classical methods in medium- or large-scale power systems are somewhat restricted.

1.1 Artificial Intelligence Methods

In the last two decades, the use of population based meta-heuristic artificial intelligence methods has been increased to solve an ORPD problem to curb the limitations of classical optimization methods. In general, these methods make use of intelligent search and probabilistic rules to find better solutions in each iteration. Lee et al. [9] proposed an application of genetic algorithm (GA) for reactive power planning in which the cost of available VAR sources and generator fuel costs were minimized. Venkatesh et al. [10] used fuzzy logic method to consider the uncertainty in information while minimizing losses and maximizing voltage security simultaneously. Zhang et al. [11] solved reactive power planning using tabu search method that uses flexible memory in obtaining global solution. Chen et al. [12] solved multi-objective optimization in which the VAR cost, total real power loss, and voltage deviation of the load buses from their nominal values had been optimized simultaneously using simulated annealing algorithm. Zhao et al. [13] suggested multi-agent based particle swarm optimization in which particles compete and cooperate with their neighbors to find the best solution. Dai et al. [14] solved ORPD problem using a seeker algorithm in which it uses the act of human searching and step length is decided by fuzzy rules. El-Ela et al. [15] used ant colony optimization algorithm and sensitivity-based approach to solve ORPD problem. Salaan and Estoperez [16] proposed a real-time application of artificial neural network (ANN) to optimize reactive power flow in the distribution system. Raha et al. [17] suggested modified differential evolution (DE) algorithm to increase convergence speed of original DE for solving ORPD problem. Saraswat and Saini [18] used fuzzy multi-objective evolutionary algorithm to solve ORPD problems in which crossover probability and mutation probability were dynamically changed using a fuzzy logic controller.

Due to the inclusion of wind farms into conventional power systems, additional constraints of WPP make ORPD problem quite nonlinear and highly complex. Up to now, the following research has been done to tackle this problem:

Villanueva et al. [19] used a probabilistic load flow method to correlate wind power with generation and load demand. Erlich and Nakawiro [20] proposed mean–variance mapping optimization to minimize the losses and cost of on-load tap changing movement in wind farms. Rojas et al. [21] suggested the application of PSO to minimize the active power losses in the wind power system.

Cuckoo search algorithm is a new meta-heuristic artificial intelligence optimization method, which has been used to solve ORPD. The obtained results have been compared with that of PSO [22]. The following sections give brief introduction about CSA.

1.2 Cuckoo Search Algorithm (CSA)

It is one of the latest nature-inspired meta-heuristic algorithms, developed in 2009 by Yang and Deb [23]. CSA is based on the brood parasitism of some cuckoo species. In addition, this algorithm is enhanced by the so-called Lèvy flights [24], rather than by simple isotropic random walks. Some species such as the *ani* and *Guira* cuckoos lay their eggs in communal nests, though they may remove others' eggs to increase the hatching probability of their own eggs [25]. Quite a number of species engage the obligate brood parasitism by laying their eggs in the nests of other host birds. For simplicity in describing the standard cuckoo search [23], we use the following three idealized rules: (I) Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest; (II) The best nests with high quality of eggs (solutions) will carry over to the next generations; and (III) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $P_a \in (0, 1)$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

This algorithm uses a balanced combination of a local random walk and the global explorative random walk, controlled by a switching parameter p_a and the local random walk can be written as [23]

$$x_i^{(t+1)} = x_i^{(t)} + \alpha s \otimes H(p_a - \varepsilon) \otimes (x_j^t - x_k^t) \tag{1}$$

where x_j^t and x_k^t are the two different solutions selected randomly by random permutation, H(u) is a Heaviside function, ε is a random number drawn from a uniform distribution, and s is the step size. On the other hand, the global random walk is carried out using Levy flights:

$$x_i^{t+1} = x_i^t + \alpha\, L(s,\lambda),$$
$$L(s,\lambda) = \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0). \tag{2}$$

Here $\alpha > 0$ is the step size scaling factor, which should be related to the scales of the problem of interest. According to [23], $\alpha = o(L/10)$ where L is the characteristic scale of the problem of interest, while in some cases $\alpha = o(L/100)$ can be more effective and avoid flying too far.

2 Wind Power Plant System

The single line diagram of a 41-bus wind power plant system is shown in Fig. 1. It is taken from [26] and some data have been modified to consider a wind power system of INDIA. The data are given in the format of MATPOWER 4.1 [27]. As shown in Fig. 1, the conventional power system is represented as a slack bus generator that is connected to bus-1. WPP is connected to the slack bus through short transmission lines, transformers, and cables. There are in total 18 doubly fed induction generators (DFIGs), each having the active power rating of 2.1 MW and reactive power rating of 1.84 MVAR. Its voltage rating is 0.690 kV. This voltage is step up to 33 kV using step-up transformer which is equipped with taps to change the voltage on the HV side. All DFIGs are finally connected to the bus-5 via short

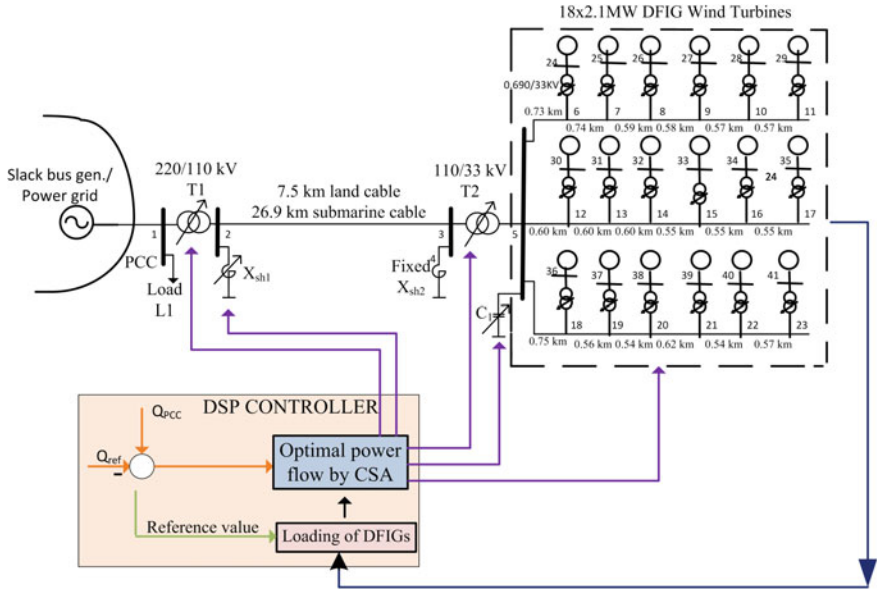

Fig. 1 Modified Wind Power Plant (WPP) system [26]

transmission lines. On the same bus, the stepwise (discrete) regulated capacitor C_1 is connected to provide auxiliary reactive power support. Transformer T_2 steps up 33–110 kV. Bus-2 and bus-3 are connected through land and submarine cables. As cables provide large charging currents, two shunt reactors are connected at buses 2 and 3. Reactor Xsh_2 provides fixed VAR compensation, whereas Xsh_1 can provide VAR compensation in a continuous manner. Transformer T_1 steps up 110–220 kV at bus-1. Both transformers T_1 and T_2 are equipped with stepwise adjustable on-load tap positions that can be changed to control reactive power flow. A factitious load L_1 is connected at bus-1, which is a point of common coupling (PCC) at which powers coming from the grid are merged with those coming from DFIGs. Load L_1 represents a reactive power requirement (Q_{ref}) at PCC which must be fulfilled by DFIGs and power grid in such way that active power losses of transmission lines and cables get minimized. So the difference between the injected reactive power (Q_{PCC}) by the DFIGs and reactive power requirement (Q_{ref}) should be very near to zero so as to minimize the total losses of the wind power system.

Assuming that the online data about the actual status of DFIGs, transformer tap positions, and compensating devices are available through the SCADA system, following optimization strategy can be formulated to minimize the losses of the wind power system.

3 Problem Formulation

The aim of optimal reactive power dispatch is to minimize the total active power losses of the transmission system while fulfilling constraints, which can be defined from the following objective:

$$\min \sum_{k \in N_E} P_{\text{kloss}} = \sum_{k \in N_E} g_k (v_i^2 + v_j^2 - 2 v_i v_j \cos \theta_{ij}) \tag{3}$$

where

$k = (i,j);\ i \in N_B$ (Total no. of buses)

$j \in N_i$ (No. of buses adjacent to bus i, including bus i)

$\sum_{k \in N_E} P_{\text{kloss}}$ = Total active power losses in the transmission system

N_E : Set of numbers of transmission lines

$g_k =$ Conductance of branch k(pu)

$v_i, v_j =$ voltage magnitude (pu) of bus i and j respectively

$\theta_{ij} =$ load angle difference between bus i and j(rad)

This is subject to the following constraints to be outlined in the next subsections.

3.1 Equality Constraints

These constraints confirm that the algebraic summation of injected active powers and algebraic summation of extracted active powers at a particular bus should be the same, i.e., their difference must be zero. Similarly, the algebraic summation of injected reactive powers and algebraic summation of extracted reactive powers at a particular bus should be the same.

Active power flow balance equations at all buses excluding the slack bus:

$$P_{gi} - P_{di} - v_i \sum_{j \in N_i} v_j (g_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) = 0 \tag{4}$$

Reactive power flow balance equations at all PQ buses (load buses):

$$Q_{gi} - Q_{di} - v_i \sum_{j \in N_i} v_j (g_{ij} \sin \theta_{ij} + B_{ij} \cos \theta_{ij}) = 0 \tag{5}$$

An additional equality constraint due to inclusion of WPP is

$$Q_{\text{PCC}} - Q_{\text{ref}} = 0 \tag{6}$$

Reactive power injected (Q_{PCC}) at the point of common coupling (PCC) should be equal to the reactive power demand (Q_{ref}) so as to maintain an acceptable voltage level at PCC as shown in Fig. 1.

3.2 Inequality Constraints

Inequality constraints ensure that the values of some variables like reactive power output of generators, wind turbines, reactors, and capacitors should be maintained between their lower bound and upper bound for the safe operation of the equipment. Voltage at each bus should be typically maintained between 0.9 and 1.1 p.u. to avoid under-voltage and Ferranti effect, respectively. The power flow in the transmission line should be less than its maximum thermal limit to avoid any congestion.

Reactive power generation limit for each generator bus

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max},\ i \in N_g \tag{7}$$

reactive power generation limit for each wind turbine

$$Q_{WTi}^{\min} \leq Q_{WTi} \leq Q_{WTi}^{\max},\ i \in N_{WT} \tag{8}$$

reactive power limits of reactors and capacitors

$$Q_{Li}^{\min} \le Q_{Li} \le Q_{Li}^{\max},\ i \in N_L \tag{9}$$

$$Q_{Ci}^{\min} \le Q_{Ci} \le Q_{Ci}^{\max},\ i \in N_C \tag{10}$$

voltage magnitude limit for each bus

$$v_i^{\min} \le v_i \le v_i^{\max},\ i \in N_B \tag{11}$$

Transformer tap-setting constraint is

$$T_k^{\min} \le T_k \le T_k^{\max} \tag{12}$$

The apparent power flow limit constraint of each transmission line is

$$s_l \le s_l^{\max}, \qquad \forall l \in N_E \tag{13}$$

The augmented objective function (fitness function) can be formulated as follows:

$$F_P = \sum_{k \in N_E} P_{\text{kloss}} + \text{Dynamic Penalty Function} \tag{14}$$

where

$$\begin{aligned}\text{Dynamic Penalty Function} &= \text{Dynamic Penalty coefficient} * \left(\sum \text{constraint violation}\right)^2 \\ &= (a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5) \\ &\quad * \left\{ \left(\sum_{i=1}^{N_G} f(Q_{gi}) \right)^2 + \left(\sum_{i=1}^{N} f(V_i) \right)^2 + \left(\sum_{m=1}^{N_E} f(S_{lm}) \right)^2 \right\}\end{aligned} \tag{15}$$

$$f(x) = \left\{ \begin{array}{lll} 0 & \text{if} & x^{\min} \le x \le x^{\max} \\ (x - x^{\max})^2 & \text{if} & x > x^{\max} \\ (x^{\min} - x)^2 & \text{if} & x < x^{\min} \end{array} \right\} \tag{16}$$

where $a_i \in [0,1]$ are user-defined constants $a_0 = 0,\ a_1, a_2,\ a_3,\ a_4, a_5 = 1$. Here, t is the iteration number.

A dynamic penalty function [28] is used to handle the constraint violations of the reactive power output of generator buses, wind turbines, reactors and capacitors, bus voltage magnitudes of all buses, and line flow limits of all transmission lines. It consists of the penalty coefficient, which is dynamically increased as the iteration

number (t) increases. So it has the property of allowing highly infeasible solutions early in the search space, while continually increasing the penalty imposed eventually move the final solution to the feasible region.

4 Step-by-Step Procedure for ORPD Using CSA for a 41-Bus Wind Power Plant System

(1) Define and initialize randomly a total of 22 control variables as given in Table 1 within their permissible ranges; define the size of host nests (n), the discovery rate of alien eggs (P_a = 0.25), number of iteration (=10,000), convergence criteria, and input the data of the 41-bus test system.
(2) Take iteration number $t = 0$.
(3) For each cuckoo's egg (nest), run Newton–Raphson (NR) load flow to find out total transmission losses.
(4) Calculate the fitness of each cuckoo's egg using Eq. (14)
(5) Find the best nest, which gives the minimum value of the objective function.
(6) $t = t + 1$.
(7) Generate new nests by performing Levy flights using Eqs. (1) and (2).
(8) Evaluate the fitness (F_i) of each new nest by performing the NR load flow analysis.
(9) Choose a nest (say, j) randomly.
(10) If ($F_i > F_j$), then replace j by the new solutions.
(11) A fraction (p_a) of worse nests is abandoned and new nests are built at new locations via Lèvy flights.
(12) Keep the best solutions obtained so far. Rank the solutions and find the current best solutions.
(13) Go to step number 6 until the convergence criteria are satisfied.
(14) Coordinates of best nest give optimized values of control variables and its fitness gives minimized value of losses.

Table 1 Control variables of the optimization problem

Sr no.	Name of control variable	Type of control variable	Quantity	Limits of variables	
				Min.	Max.
1	Reactive power output of DFIGs (MVAR)	Continuous	18 (As there are total 18 DFIGs)	−1.84	+1.84
2	Tap positions of Transformers	Discrete	02 (each for T_1 and T_2)	−5	+9
3	Reactive power absorption of X_{sh1}	Continuous	01	0	1
4	Stepwise adjustment of capacitor C_1	Discrete	01	0	5
Total control variables			22	–	

5 Simulation Results and Discussion

Simulation studies were carried out on a computer with an Intel Xeon CPU, 2.67-GHz system (dual processors), 5 GB of RAM, MATLAB 7.10 platform [The MathWorks, Natick, Massachusetts, USA]. The 41-bus WPP is used to test the effectiveness and efficiency of CSA. The performance of CSA has been compared with PSO. The developed MATLAB codes can be obtained from the authors.

The reactive power requirement (Q_{ref}) at PCC is 0.2 p.u., which should be fulfilled by the WPP system. However, the load flow analysis at the base case yields that all bus voltages (except buses 1, 2, 3, and 4) violate their permissible upper limits of 1.10 p.u. This is because of large mismatch between the reactive power requirement and reactive power injection at PCC. Also, active and reactive output powers of all DFIGs hit their maximum limits of 2.1 MW and 1.84 MVAR, respectively. So the load flow simulation has infeasible solutions and so it is concluded that it is a hard-to-solve optimization problem. CSA has been tested to solve such a complex problem.

The control variables and their permissible limits are given in Table 1. It is seen that the optimization problem consists of continuous and discrete variables. So it is a nonlinear mixed integer optimization problem. The total number of constraints is 123, which include the equality constraints of power flow balance equations at each bus, difference between Q_{PCC} and Q_{ref}, and inequality constraints of bus voltage limits and line power flow limits. Load flow data are given in Table 2.

Table 3 shows the optimized results of these control variables obtained by PSO and CSA methods. Some values of the reactive output powers of wind turbine DFIGs are positive, which indicates that they supply reactive powers to the grid to fulfill the reactive power demand of load L_1, whereas their negative values indicate that DFIGs absorb reactive powers from the grid to maintain the acceptable voltage profile of the buses. $T_{1–2}$ and $T_{3–5}$ consist of total 15 discrete tap positions. Tap number +2 indicates the nominal HV side voltages, i.e., 220 kV and 110 kV for $T_{1–2}$ and $T_{3–5,}$ respectively. Tap positions from 3–9 will decrease the HV side voltage and tap position from (1 to −5) will increase HV side voltage in a discrete step of 1.25 %. of nominal voltage. This facilitates the voltage variation of about −10 to +10 % at buses 1 and 3. Shunt reactor at bus 2 absorbs and capacitor C_1 supplies reactive powers in the system.

As shown in Table 4, the total active power losses obtained by CSA are 15.56 % lesser than that of PSO. It confirms that CSA outperforms PSO in terms of the quality of solutions. Such a high reduction in losses by CSA will decrease the cost of losses and ensures economical operation of the power system. CSA also obtains the lesser absolute value of reactive power mismatch (ΔQ_{PCC}) than PSO, which indicates that CSA can easily fulfill an additional constraint number (6) of WPP with a higher accuracy.

As PSO and CSA are stochastic optimization methods, the initial populations are generated randomly and, as a consequence, they give slightly different optimized results in every simulation. So a total of 31 independent simulations were carried

Table 2 Load flow data in MATPOWER format

Bus data							
Bus no.	*Pd*	*Qd*	*Gs*	*Bs*	baseKV	Vmax (pu)	Vmin (pu)
1	0	0	0	0	220	1.1	0.9
2	0	0	0.5208	−12.1	110		
3	0	0	0	0	110		
4	0	0	0.4586	−8.0667	110		
5	0	0	0	9.8965	33		
6	0	0	0.0049	0			
7	0	0	0.0049	0			
8	0	0	0.0049	0			
9	0	0	0.0049	0			
10	0	0	0.0049	0			
11	0	0	0.0049	0			
12	0	0	0.0049	0			
13	0	0	0.0049	0			
14	0	0	0.0049	0			
15	0	0	0.0049	0			
16	0	0	0.0049	0			
17	0	0	0.0049	0			
18	0	0	0.0049	0			
19	0	0	0.0049	0			
20	0	0	0.0049	0			
21	0	0	0.0049	0			
22	0	0	0.0049	0			
23	0	0	0.0049	0			
24	−2.1	−1.84	0	0	0.69	1.05	0.95
25	−2.1	−1.84	0	0			
26	−2.1	−1.84	0	0			
27	−2.1	−1.84	0	0			
28	−2.1	−1.84	0	0			
29	−2.1	−1.84	0	0			
30	−2.1	−1.84	0	0			
31	−2.1	−1.84	0	0			
32	−2.1	−1.84	0	0			
33	−2.1	−1.84	0	0			
34	−2.1	−1.84	0	0			
35	−2.1	−1.84	0	0			
36	−2.1	−1.84	0	0			
37	−2.1	−1.84	0	0			
38	−2.1	−1.84	0	0			
39	−2.1	−1.84	0	0			
40	−2.1	−1.84	0	0			
41	−2.1	−1.84	0	0			

(continued)

Table 2 (continued)

Bus data							
Generator data							
Bus	*Pg*	*Qg*	*Q*max	*Q*min	*P*max	*P*min	
1	0	0	1000	−1000	800	0	
Branch data							
fbus	tbus	r	x	b	rateA	ratiomax	ratiomin
1	2	0.0016	0.064	0	200	1.149	0.851
2	3	0.0024	0.0104	0.06536	77.11	0	0
3	4	0.0136	0.0241	0.2115	77.11	0	0
4	5	0.032	0.1654	0	100	1.13	0.87
6	24	0.0065	1.5282	0	2.5	0	0
7	25	0.0065	1.5282	0	2.5	0	0
8	26	0.0065	1.5282	0	2.5	0	0
9	27	0.0065	1.5282	0	2.5	0	0
10	28	0.0065	1.5282	0	2.5	0	0
11	29	0.0065	1.5282	0	2.5	0	0
12	30	0.0065	1.5282	0	2.5	0	0
13	31	0.0065	1.5282	0	2.5	0	0
14	32	0.0065	1.5282	0	2.5	0	0
15	33	0.0065	1.5282	0	2.5	0	0
16	34	0.0065	1.5282	0	2.5	0	0
17	35	0.0065	1.5282	0	2.5	0	0
18	36	0.0065	1.5282	0	2.5	0	0
19	37	0.0065	1.5282	0	2.5	0	0
20	38	0.0065	1.5282	0	2.5	0	0
21	39	0.0065	1.5282	0	2.5	0	0
22	40	0.0065	1.5282	0	2.5	0	0
23	41	0.0065	1.5282	0	2.5	0	0
5	6	0.0081	0.0279	0.0024	18.3	0	0
6	7	0.0023	0.0064	4.90e-04	16.5	0	0
7	8	0.0022	0.0062	4.73e-04	16.5	0	0
8	9	0.0052	0.008	4.03e-04	11.7	0	0
9	10	0.0046	0.007	3.51e-04	11.7	0	0
10	11	0.0048	0.0074	3.70e-04	11.7	0	0
5	12	0.0019	0.0067	5.76e-04	18.3	0	0
12	13	0.0025	0.0069	5.25e-04	16.5	0	0
13	14	0.0025	0.0069	5.25e-04	16.5	0	0
14	15	0.0046	0.0071	3.57e-04	11.7	0	0
15	16	0.0046	0.0071	3.57e-04	11.7	0	0

(continued)

Table 2 (continued)

Bus data							
16	17	0.0046	0.0071	3.57e-04	11.7	0	0
5	18	0.0056	0.0193	0.0017	18.3	0	0
18	19	0.003	0.0085	6.48e-04	16.5	0	0
19	20	0.0024	0.0068	5.14e-04	16.5	0	0
20	21	0.0049	0.0075	3.77e-04	11.7	0	0
21	22	0.0048	0.0074	3.70e-04	11.7	0	0
22	23	0.0048	0.0074	3.70e-04	11.7	0	0
Profile of active power (MW) dispatch for individual wind turbines							
1.9326	1.83	1.7302	1.7878	1.8567	1.6	1.8143	1.7906
1.1	1.8244	1.8278	1.2	1.325	1.74	1.3583	1.8263
1.256	1.8286						
Profile of WPP reactive power (MVAR) requirement at PCC							
0.2							

Table 3 Optimized values of control variables by PSO and CSA

Bus no.	Control variables	PSO	CSA
24	Q_{WT24} (MVAR)	1.2387	−0.8344
25	Q_{WT25}	−1.3022	−0.9453
26	Q_{WT26}	0.1164	−0.7528
27	Q_{WT27}	−0.9368	1.5943
28	Q_{WT28}	−0.9562	0.9287
29	Q_{WT29}	−0.5498	−1.4870
30	Q_{WT30}	0.1676	−1.4363
31	Q_{WT31}	1.0119	−1.6400
32	Q_{WT32}	−1.5894	0.1722
33	Q_{WT33}	−0.3419	−1.6500
34	Q_{WT34}	0.5448	0.5943
35	Q_{WT35}	−0.3629	1.5179
36	Q_{WT36}	1.1042	−0.6667
37	Q_{WT37}	−1.0069	0.4258
38	Q_{WT38}	−0.2083	0.7845
39	Q_{WT39}	−0.1325	−1.6500
40	Q_{WT40}	1.3862	−0.5946
41	Q_{WT41}	−0.8374	−0.5770
1–2	T_{1-2} (p.u.)	9	9
3–5	T_{3-5}	−2	−2
2	X_{sh1}	0.6103	00
5	C_1	5	1

out for both the methods and obtained losses are tabulated in Table 5. The average (mean) value of losses obtained from CSA is quite lower than PSO. Also, the frequency of achieving losses lesser than mean value is 13 for PSO and 20 for CSA, which concludes that CSA is more robust than PSO in achieving the global optimum solution.

Table 4 Comparisons of the optimized results of PSO and CSA

	PSO	CSA	$\%P_{SAVE}$ by CSA
Active power losses (MW)	1.0157	0.8576	15.56
ΔQ_{PCC}(pu)	−0.0489	0.0168	–

Table 5 Statistical analysis of PSO and CSA over 31 independent runs

Algorithm	Best losses (MW)	Worst losses (MW)	Mean losses (MW)	F^*
PSO	1.0157	1.1367	1.0783	13
CSA	0.8576	1.2814	0.9486	20

F^* Frequency of achieving losses, lesser than the mean value

Table 6 Computing time for PSO and CSA

Algorithm	Shortest time (s)	Longest time (s)	Average time (s)
PSO	74.22	75.75	74.81
CSA	72.93	76.51	75.30

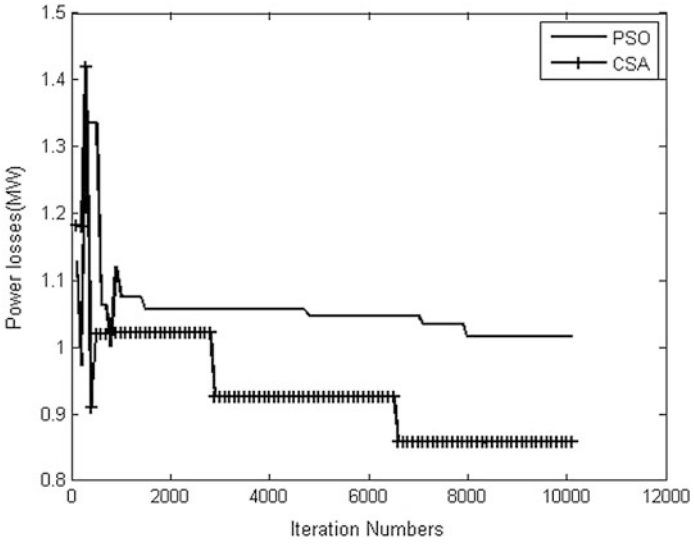

Fig. 2 Convergence graphs of PSO and CSA

Table 7 Selected parameters of PSO and CSA

Sr no.	Parameters	PSO	CSA
1	Population size (n)	100	05
2	Acceleration constant (C1, C2)	2.1 and 2.0	NA
3	Constriction factor	0.729	NA
4	Max. and Min. inertia weights	1 and 0.2	NA
5	Max. and Min. velocity of particles	0.003 and −0.003	NA
6	Discovery rate of alien eggs (P_a)	NA	0.25
7	Convergence criteria	10,000 iterations or $\Delta f = 0.00001$	

The simulation times of both methods are depicted in Table 6. It is seen that CSA is faster than PSO in obtaining global solutions.

Figure 2 shows the convergence characteristic of both methods by considering the parameters given in Table 7. The algorithm stops when the variations of the objective function values (Δf) are less than a given tolerance of 0.00001. It can be seen that CSA has fewer parameters to be adjusted as compared to PSO. There is essentially only a single parameter P_a in CSA (apart from the population size n). So CSA is very easy to implement.

6 Conclusion

In this chapter, a CSA-based algorithm with Lèvy flights has been suggested to solve an ORPD problem of a WPP system. A comparative study reveled that CSA outperformed PSO in terms of the quality of solutions and execution time. It is because PSO can converge quickly to the current best solution, but not necessarily the global best solutions. In fact, PSO updating equations do not satisfy the global convergence conditions, and thus there is no guarantee for global convergence. On the other hand, it has been proved that cuckoo search satisfies the global convergence requirements and thus has guaranteed global convergence properties. This implies that for multimodal optimization, PSO may converge prematurely to a local optimum, while cuckoo search can usually converge to the global optimum.

Furthermore, cuckoo search has two search capabilities: local search and global search, controlled by a switching/discovery probability. Consequently, the global optimality can be found with a higher probability. It is hoped that this powerful algorithm may be extended to solve multi-objective OPF problems and may be hybridized with other methods to further increase its effectiveness.

References

1. Indian Electricity Grid Code. http://www.cercind.gov.in. pp. 1–73
2. Deeb, N., Shaidepour, S.M.: Linear reactive power optimization in a large power network using the decomposition approach. IEEE Trans. Power Syst. **5**(2), 428–435 (1990)
3. Gomez, T., Perez-Arriaga, I.J., Lumbreras, J., Parra, V.M.: A security-constrained decomposition approach to optimal reactive power planning. IEEE Trans. Power Syst. **6**(3), 1069–1076 (1991)
4. Wu, Y.C., Debs, A.S., Marsten, R.E.: A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows. IEEE Trans. Power Syst. **9**(2), 876–883 (1994)
5. Kermanshahi, B., Takahashi, K., Zhou, Y.: Optimal operation and allocation of reactive power resource considering static voltage stability. In: Proceedings of the International Conference on Power System Technology, vol. 2, pp. 1473–1477, Aug 1998
6. Chattopadhyay, D., Chakrabarti, B.B.: Voltage stability constrained VAR planning: Model simplification using statistical approximation. Int. J. Elect. Power Energy Syst. **23**(5), 349–358 (2001)
7. Yan, W.,Yu, J., Yu, D.C., Bhattarai, K.: A new optimal reactive power flow model in rectangular form and its solution by predictor corrector primal dual interior point method. IEEE Trans. Power Syst. **21**(1), 61–67 (2006)
8. Rabiee, A., Vanouni, M., Parniani, M.: Optimal reactive power dispatch for improving voltage stability margin using a local voltage stability index. Energy Convers. Manag. **59**, 66–73 (2012)
9. Lee, K.Y., Bai, X., Park, Y.M.: Optimization method for reactive power planning by using a modified simple genetic algorithm. IEEE Trans. Power Syst. **10**(4), 1843–1850 (1995)
10. Venkatesh, B., Sadasivam, G., Khan, M.A.: An efficient multi-objective fuzzy logic based successive LP method for optimal reactive power planning. Elect. Power Syst. Res. **59**(2), 89–102 (2001)
11. Zhang, W., Liu, Y., Liu, Y.: Optimal VAR planning in area power system. In: Proceedings of the International Conference on Power System Technology, pp. 2072–2075, 13–17 Oct 2002
12. Chen, Y.L., Ke, Y.L.: Multi-objective VAR planning for large-scale power systems using projection based two-layer simulated annealing algorithms. IEEE Proc. Gener. Transm. Distrib. **151**(4), 555–560 (2004)
13. Zhao, B., Guo, C., Cao, Y.: A Multi-agent based particle swarm optimization approach for optimal reactive power dispatch. IEEE Trans. Power Syst. **20**(2), 1070–1078 (2005)
14. Dai, C., Chen, W., Zhu, Y., Zhang, X.: Seeker optimization algorithm for optimal reactive power dispatch. IEEE Trans. Power Syst. **24**(3), 1218–1231 (2009)
15. El-Ela, A.A., Kinawy, A.M., El-Sehiemy, R., Mouwafi, M.: Optimal reactive power dispatch using Ant colony optimization algorithm. Electr. Eng. **93**(2), 103–106 (2011)
16. Salaan, C.J., Estoperez, N.R.: An artificial neural network based real-time reactive power controller. In: Proceedings of the World Congress on Engineering and Computer Science (WCECS-2011), San Francisco, USA, Oct 2011
17. Raha, S.B., Chakraborty, N.: Tuned reactive power dispatch through modified differential evolution technique. Front. Energy **6**(2), 138–147 (2012)
18. Saraswat, A., Saini, A.: Multi-objective optimal reactive power dispatch considering voltage stability in power systems using HFMOEA. Eng. Appl. Artif. Intell. **26**(1), 390–404 (2013)
19. Villanueva, D., Feijoo, A., Pazos, J.: Probabilistic load flow considering correlation between generation, loads and wind power. Smart Grid Renew. Energy **2**, 12–20 (2011)
20. Erlich, I., Nakawiro, W.: Optimal dispatch of reactive sources in wind farms. Presented at Power and Energy Society General Meeting, pp. 1–7, 24–29 July 2011
21. Rojas, M., Sumper, M., Bellmunt, A., Andreu, O.: Reactive power dispatch in wind farms using particle swarm optimization techniques and feasible solutions search. Appl. Energy **88** (12), 4678–4686 (2011)

22. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
23. Yang, X.S., Deb, S.: Cuckoo search via Levy flights. In: Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), IEEE publications, pp. 210–214 (2009)
24. Pavlyukevich, I.: Levy flights, non-local search and simulated annealing. J. Comput. Phys. **226**, 1830–1844 (2007)
25. Payne, R.B., Sorenson, M.D., Klitz, K.: The Cuckoos. Oxford University Press, Oxford (2005)
26. Erlich, I., Lee, K.Y., Rueda, J.L., Wildenhues, S.: IEEE PES competition on application of modern heuristic optimization algorithms for solving optimal power flow problems: Problem definitions and implementation guidelines. p. 4 https://www.unidue.de/ean/downloads/aktuelles/Invitation_letter.pdf. Sept 2013
27. Zimmerman, R.D., et al.: MATPOWER: Steady-State Operations, planning and analysis tools for power systems research and education. IEEE Trans. on Power Syst. **26**(1), 12–19 (2011)
28. Joines, J.A., Houck, C.R.: On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA. In: Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 579–584 (1994)

A DSS-Based Honeybee Mating Optimization (HBMO) Algorithm for Single- and Multi-objective Design of Water Distribution Networks

Omid Bozorg Haddad, Navid Ghajarnia, Mohammad Solgi, Hugo A. Loáiciga and Miguel Mariño

Abstract A decision support system (DSS) for long-term design of water distribution networks (WDNs) herein named "dynamic design" is proposed in this work. The proposed DSS is capable of recognizing the long-term consequences of various WDN initial designs to achieve a desirable performance in the rehabilitation period. Single- and multi-objective initial designs and rehabilitation problems are considered in which the design variables are the pipes' diameters and several rehabilitation alternatives. The DSS relies on the honeybee mating optimization (HBMO) and the multi-objective honeybee mating optimization (MOHBMO) algorithms to minimize the total cost of the initial implementation of a WDN and of its rehabilitation cost, and/or maximize the WDN's hydraulic reliability. This paper's results show the advantages of a DSS that considers design and rehabilitation (dynamic design) of activities simultaneously in comparison to DSSs that minimize the initial cost of WDNs only (normal design).

Keywords Decision support system · Single-objective optimization · Multi-objective optimization · Honeybee mating optimization algorithm · Initial design · Rehabilitation · Water distribution system

O. Bozorg Haddad (✉) · N. Ghajarnia · M. Solgi
University of Tehran, Tehran, Iran
e-mail: obhaddad@ut.ac.ir

H.A. Loáiciga
University of California, Santa Barbara, Santa Barbara, CA, USA

M. Mariño
University of California, Davis, Davis, CA, USA

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*,
Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_10

1 Introduction

A decision support system (DSS) is a computer-based information system that supports decision-making activities. DSSs serve management, operations, modeling, and planning tasks and help in making decisions, specially when there may be rapidly changing conditions that affect the function of a knowledge-based system. A properly designed DSS is an interactive software-based system intended to help decision makers compile useful information from a combination of input data, knowledge, predefined scenarios, or conceptual models to identify, analyze, solve problems, and make decisions.

According to Keen and Marton [44], the concept of decision support has evolved from two main areas of research: The theoretical studies of organizational decision making done at the Carnegie Institute of Technology during the late 1950s and early 1960s, and the technical work on interactive computer systems, were mainly carried out at the Massachusetts Institute of Technology in the 1960s. It is considered that the concept of DSS became an area of research of its own in the mid 1970s.

In 1987, Sol et al. [68] remarked that the definition and scope of DSSs have been migrating over the years. In the 1970s, a DSS was described as "a computer based system to aid decision making." In the late 1970s, the DSS movement started focusing on "interactive computer-based systems which help decision-makers utilize data bases and models to solve ill-structured problems." In the 1980s, DSSs were thought of as providing systems "using suitable and available technology to improve effectiveness of managerial and professional activities," and at the end of the 1980s DSSs faced a new challenge concerning design of intelligent workstations.

Nowadays, a DSS is usually used for integrated analyzing and planning of water systems during their initial design and operational periods. Since different circumstances affecting water systems change endlessly, DSSs can help decision makers a great deal in water system analysis and planning during the operational period. DSSs introduce optimal operational choices intended to meet suitable conditions from the viewpoint of consumer satisfaction, water system sustainable development, and also economic considerations by introducing input data, defining water system feedbacks through mathematical functions, and benefiting from different operational policies for the system in an interactive mode.

Water distribution networks (WDNs) are one of the most important, costly, and vital infrastructures which exist in most urban areas. A huge amount of capital is spent on the design of new water distribution networks (WDNs) and the rehabilitation of existing ones in developing and developed countries. These large capital costs encourage further research on the optimal design of WDNs considering cost and hydraulic benefits. It is possible to optimize a design-operation WDN project from different perspectives using DSSs during the initial design and operation of such systems.

Numerous researches have been conducted in many fields of water resource systems such as reservoir operation (e.g., [26, 27, 29]), hydrology [55], project management (e.g., [16, 28]), cultivation rules (e.g., [14, 30]), pumping scheduling [17], hydraulic structures [15], water distribution networks (e.g., [63, 64]), operation of aquifer systems [8], site selection of infrastructures [43], and algorithmic developments [67]. However, only a few of these works dealt with the developing application of the DSSs for long-term design of water distribution networks.

Any WDN generally consists of consumption nodes with known water demands, linking pipes, pumping stations, valves, and storage reservoirs or tanks. WDNs are designed for a predefined service life. Because nodal demands of WDNs usually increase during the operational period, demand values corresponding to the final year of their useful life are considered in the initial design of a WDN. This increase can be predicted using various prediction methods. Any WDN is continuously subjected to environmental and operational stresses that lead to its deterioration [39]. On the other hand, because of the decay (say corrosion) of a WDN's components (mainly pipes) or unexpected accidents (such as pipe breakages), almost all WDNs need rehabilitation during their useful life. Therefore, after initial design of any WDN, rehabilitation of old components, if they exist, is needed. Improvement of WDN design projects is possible by means of integrated DSS for their initial and rehabilitation design. To better support this assertion a brief state of the art regarding WDN designs is presented below.

WDNs were in past decades designed using trial-and-error methods by experienced engineers. Since these methods were typically dependent on human experience and judgement and on trial-and-error attempts, optimal solution of the WDN design problem was not expected given that it was achieved subjectively and relying on trial-and-error methods. Several researchers have focused on the optimal design of WDNs using linear programming (e.g., [3, 36, 45, 57]) and nonlinear programming [31].

In spite of achieving improvements in WDN design using LP and NLP, Savic and Walters in 1997 [61] pointed out that the hydraulic constraints of the problem must be checked to adjust the continuous pipe size solutions to commercially available ones. This is so because pipe diameters in LP and NLP methods were considered to be continuous while in practical problems the designer has to choose optimal diameters among commercially available diameters. As a result, final solutions of LP and NLP methods had to be modified to be applicable in real problems. Cunha and Sousa in 2001 [19] stated that conversion of the solutions into commercial pipe diameters could decrease the optimality of the answers and might lead to the violation of hydraulic constraints. Evolutionary optimization algorithms (EAs) have emerged as an attractive alternative for engineering design, including the problem of WDN design.

Savic and Walters in 1997 [61] applied the evolutionary-based genetic algorithm (GA) to solve the problem of WDN design. They showed that evolutionary algorithms overcome some of the shortcomings of LP and NLP. Evolutionary and metaheuristic methods are efficient tools to solve nonlinear problems and do not require linearization or calculation of partial derivatives. Investigators such as

Cunha and Sousa in 1999 [20], Lippai et al. in 1999 [50], Eusuff and Lansey in 2003 [24], Maier et al. in 2003 [51], Geem in 2005 [32], and Suribabu and Neelakantan in 2006 [73] reported the application of Simulated Annealing (SA), Taboo Search (TS), Shuffled Frog Leaping Algorithm (SFLA), Ant Colony Optimization (ACO), Harmonic Search (HS), and Particle Swarm Optimization (PSO), respectively, in the optimal design of WDNs. These algorithms have reported improved designs for WDNs in comparison with those calculated with traditional mathematical methods. A few researchers have exploited EAs and metaheuristic algorithms to solve problems of WDN calibration [60], operation (e.g., [69, 70]), and simulation [71].

All the cited studies aimed at minimizing WDN capital costs. In this regard, Walski in 2001 [76] lamented that in spite of numerous papers on the optimal design of WDN in prior decades, none had managed successfully to play an important role in the design of real WDNs. He claimed that excessive attention was devoted to minimizing the cost of WDNs and insufficient attention was given to their efficiency, this being the main reason why optimization methods for WDN have largely remained confined to the theoretical realm. Over reliance on cost minimization of WDS overlooks efficiency factors such as the reliability in meeting water demand with a desired pressure. Thus, the optimal design of WDNs must consider their cost and the reliability of their service.

Explicit consideration of reliability measures in WDN design is one of the most difficult tasks faced by designers [35]. Todini in 2000 [74] introduced a resilience index which is a surrogate for reliability and robustness of water supply. Todini in 2000 [74] applied a multi-objective design method for WDNs that minimized the WDN cost and maximized a reliability index (I_r). Afterward, Prasad and Park in 2004 [56] used the nondominant sorting genetic algorithm (NSGA) for the multi-objective design of water networks applying modifications on the I_r index. The latter authors proposed a modified reliability index named network resiliency (I_n), which improved the network's reliability in comparison with the I_r index.

The above-mentioned studies focused on initial design of WDNs, while rehabilitation projects should also be considered during the operational period of WDNs. The issue of WDN rehabilitation has been the subject of many investigations over the past 20 years. It is apparent that earlier works dealt primarily with analytical solutions to single pipe problems (e.g., [18, 22, 65, 78, 77]), while later works considered a more comprehensive network-wide approach (e.g., [5, 40–42, 46–49, 52, 62–64, 72]).

A few studies have addressed the issue of timing pipe rehabilitation alternatives in the framework of a long-term rehabilitation plan while considering structural integrity or hydraulic capacity (e.g., [6, 10, 37, 42, 52, 75]) of a WDN.

Engelhardt et al. in 2000 [23] highlighted the requirements of a rehabilitation strategy in a comprehensive review. Dandy and Engelhardt in 2006 [21] used the genetic algorithm to generate a trade-off surface that represents the compromise between cost and reliability in the problem of WDN pipe replacement. Alvisi and Franchini in 2009 [4] introduced a multi-objective genetic algorithm-based procedure for optimal medium- and short-term scheduling of leakage detection and pipe

replacement interventions in a water distribution system. Nafi et al. in 2008 [54] presented a decision support system that ensures the scheduling of pipe renewal according to available financial resources using a multi-objective approach based on Pareto ranking and the modified genetic algorithm. Nafi and Kleiner in 2010 [53] employed the same multi-objective genetic algorithm scheme, and focused on low-level scheduling of individual water mains.

Roshani and Filion in 2012 [58] also focused on the optimal scheduling of WDN rehabilitations using a multi-objective optimization framework. Their method included the broadest set of rehabilitation methods and reduced the number of decisions made by an optimization engine. Asset management strategies (including adjacency to infrastructure works, economies of scale, and annual budgetary constraints) were considered in the model and applied to the Amherstview (Ontario, Canada) water distribution system. They indicated that consideration of budgetary constraints could have negative impacts on cost while applying asset management strategies could significantly affect rehabilitation decisions and reduce cost.

The importance of the initial and rehabilitation design problems of WDNs has prompted investigators to focus deeply on each area separately; yet, it is important to consider the influence of initial network design on future rehabilitation. The availability of useful models for predicting changes in network efficiency factors (e.g., models for prediction of pipe breakage, roughness, and increasing rate of water demand) and the ability of current computer models to execute long-term simulation of WDNs enable the design and operation of WDNs with DSS.

This chapter explores the application of honeybee mating optimization (HBMO) algorithm and its multi-objective version (MOHBMO) to the problem of WDN design. The WDN design problem is the determination of optimal pipe diameters that minimizes its total capital costs of initial design and its rehabilitation. The chapter focuses on the application of a new methodology named dynamic design, to pipe sizing simultaneously with rehabilitation problems, while most previous investigations dealt with only one of those, separately. Herein, the network economics and hydraulic capacity are analyzed simultaneously over a predefined operation period, while explicitly considering the deterioration over time of the structural integrity and hydraulic capacity of every pipe in the system. This approach leads to a DSS for the simultaneous initial design and rehabilitation of WDNs. The DSS herein developed for dynamic design of WDNs calculates cheaper and more reliable solutions for decision makers to those solutions that address only cost minimization of the initial WDN design.

As mentioned earlier, a DSS mainly consists of: (1) input data; (2) a modeling software defining interactions between different system parts or components; and (3) a decision policy or logic in order to obtain an optimal plan for system operation or design. Input data such as nodal demands and pipes' roughness during the useful life of each network, nodal and reservoir elevations positions, pipe lengths, commercially available diameters for pipe and their relevant unit costs, and any other needed data are given in a database linked to the dynamic design DSS proposed herein. WDN system interactions are modeled using a hydraulic simulation program (EPANET 2.0) that computes the hydraulic variables of different network

parts (such as nodal pressures or pipe velocities) which are influenced by different initial and future conditions (such as initial and future nodal demands or pipe roughness values or initial pipe diameters). EPANET is linked with a single-/multi-objective HBMO algorithm. The main objective of the proposed DSS is to design network pipes' diameters in the initial design and in the rehabilitation period of the WDN. Subsequent parts of this chapter discuss different components of the proposed DSS.

2 Main DSS Planning Goal: Dynamic Design of Water Distribution Networks

The main planning goal of the DSS used in this chapter, named dynamic design method, is aimed at optimally designing a WDN considering the initial design and rehabilitation through its service life simultaneously. First, several definitions of the rehabilitation model are presented. Basic concepts of the dynamic design approach are then discussed. The long-term hydraulic behavior of the network is simulated by estimating the nodal demand increase of water demand and the Hazen–Williams coefficient decrease during the operation period. Let Ni and T be the number of pipes in the network and the number of operational years (including the year of installation and years of operation), respectively. The following formula expresses the number of decision variables in the optimization model in terms of the number of rehabilitation alternatives:

$$\text{NDecvar} = Ni \times (T - 1) \times \text{NAlter} \tag{1}$$

in which NAlter = the number of rehabilitation alternatives and NDecvar = the number of decision variables. Because rehabilitation does not commonly arise in the 1st year of installation, the number of operational years in Eq. (1) is reduced by one.

The DSS discussed in this chapter can choose one of the several rehabilitation alternatives in each year of operation (except the 1st year of installation) and for each link of a WDM. Hallhal in 1999 [37] introduced different rehabilitation alternatives to increase the hydraulic capacity of a WDN such as replacing and duplicating. Thus, three different options are considered as rehabilitation alternatives: (1) replacing an existing pipe with a new one having a commercially available diameter; (2) duplicating (adding) the existing link by adding a new parallel pipe; and (3) doing nothing (DoN) with the network in that situation. Due to the considered objective function and hydraulic constraints, the DSS identifies one of the rehabilitation alternatives to keep the nodal pressures above the predefined minimum permissible value; otherwise it will choose the do-nothing (DoN) option. The following practical constraints are also considered so as to make the policy of the DSS realistic and practical.

1. Implementation of any rehabilitation alternative during a few years after the start of the operation period ($\beta 1$), and during a few years before the end of the operation period ($\beta 2$) are not allowed. The reason is that a WDN must operate correctly more than a minimum of time. Moreover, it is not logical to rehabilitate a network too close to the end of its service life unless one is planning to expand the network for future years. Nafi and Kleiner in 2010 [53] have stated that the replacement cost for pipes decreases with increasing age of pipes. Therefore, it is not cost effective to replace pipes prematurely. On the other hand, the risk of pipe failures increases with increasing age. Therefore, the selection of values $\beta 1$ and $\beta 2$ is done based on the available budget, discount rate, allowable risk for pipe failures, diameter of the pipe, pipe material, etc. In this chapter $\beta 1$ and $\beta 2$ are assumed to be 10 and 2 years, respectively.
2. Halhal et al. in 1999 [37] contended that it is likely that there will be a limitation on available budget to modify or add a number of components in a water network at a specific time. Therefore, the number of rehabilitated pipes in each year should not exceed a percentage of the total number of pipes installed during the construction of the WDN. It is assumed that only a percentage of the pipes ($\beta 3$ %) can be rehabilitated at once. The value of β_3 must be assumed based on real social and practical conditions. This parameter depends on funds available in each year. In this chapter it is assumed to equal 25 %.
3. After rehabilitation of each pipe it is not possible to implement any other rehabilitation alternative on it again over a number of years following the rehabilitation. β_4 is the number of years after each rehabilitation so that during β_4 it is not possible to again rehabilitate the pipe. In other words, each rehabilitation activity has to ensure desirable performance of the network for at least β_4 years unless an unexpected event happens. The value of β_4 is determined in the same manner as β_1. In this chapter this parameter is considered for 10 years.
4. The total number of rehabilitation activities implemented during the operation period in the network should not exceed a maximum value equal to β_5. This is assumed based on available financial resources.

Capital costs of the network during the operation period consist of: (1) cost of purchasing the pipes; (2) cost of excavation and installation; and (3) scrap value (the worth of a pipe when the pipe is deemed no longer usable). In this study, the service life of pipes is considered to be equal to the service life of the network. Therefore, if the useful life of the network is, say, 30 years, then each pipe has a service life equal to 30 years after it is installed.

WDN design can be divided into three main categories: (1) initial design; (2) expansion; and (3) rehabilitation. Undoubtedly, the initial design of a network influences its hydraulic performance in subsequent years. Thus, some designers consider relatively large pipes in the initial design in order to attain a more reliable condition over the design horizon. However, the existence of large pipes in the network during its initial years of operation when nodal water demands and flow velocities are low may lead to a substantial increase in water pressure, and, consequently, to an undesirable increase of leakage and breakage rate. In addition, large

pipes are also associated with water quality problems as the residence time of water in the pipes being relatively long in this case. For this reason it is wise to save construction cost in the initial construction phase using smaller pipes and expand the WDN over time as needed.

The main goal of a dynamic design strategy is to couple rehabilitation and initial design. Therefore, dynamic design of the network can be considered as the simultaneous initial and rehabilitation design of the network. According to this method a WDN is designed to meet services under present conditions and be expandable to meet future conditions. Thus, the water demand and minimum allowable pressure constraints are satisfied with respect to the current hydraulic and structural conditions. The WDN is expanded over time incrementally as conditions might require it.

The difference between dynamic and normal design of a network can be explained by paying attention to the designer viewpoints. In normal design, the network is initially designed without considering the network's condition in rehabilitation periods. Therefore, the decisions about initial and rehabilitation designs are made separately. In contrast, the dynamic design method predicts the network's condition during rehabilitation periods, and initially designs the network in a way that facilitate rehabilitation in the future This is done by developing an optimization algorithm that chooses optimal decision variables for initial diameters and rehabilitation activities simultaneously. In contrast, the optimal choices for initial design and rehabilitation decision variables are chosen separately by separate optimization problems in normal design.

Decrease of capital costs and the increase of the network reliability are advantages of the dynamic design policy as compared to normal designs based on the predefined characteristic of the network in the initial year of installation. Postponing installation costs, has a positive effect on the economic calculations of the project due to the influence that the interest rate has on the cost stream associated with dynamic design. The dynamic design reduces capital costs in the initial design (which is often based on condition of the final year of the design horizon) and uses the savings for network expansion as conditions required in future years. Moreover, dynamic design of the network prevents an undesirable increase of water pressure using smaller pipes in the initial design and produces pipe networks with superior levels of hydraulic reliability.

3 The Optimization Problem

Two objective functions are considered: (1) minimization of the total cost associated with initial installation and rehabilitation, and (2) maximization of hydraulic reliability of the pipe network.

The total cost of the network (NetCost) consists the cost of the initial installation and rehabilitation costs. The salvage value of a pipe is taken into account, also. The second objective maximizes hydraulic reliability, and is evaluated based on the

fuzzy reliability index of the pipe network (*FRI*). The value of the *FRI* is related to the pressures on the nodes of the network so that the best value of the *FRI* is achieved while all the nodal pressures are equal to the average of the minimum and maximum allowable pressures for all times. Also, each node has different importance based on its water demand and on the diameter of the pipes that are connected to the node when evaluating the *FRI*.

Decision variables are the pipe diameters for the initial design and the respective rehabilitation times of all pipes with various rehabilitation alternatives. The constraints are: (1) minimum nodal pressure and (2) rehabilitation activity's practical considerations.

To compare the dynamic design method with the normal design method the network is initially designed considering the nodal water demand of the final year of the design horizon, using the first objective function. Afterward, the hydraulic behavior of the network is simulated considering the increases in the nodal water demand and roughness coefficient during WDN simulation. Rehabilitation is necessary due to the usual alterations of the system overt time. WDN rehabilitation is accomplished using the two objective functions. These two objective functions are also considered in the dynamic design of the network. The results from the dynamic design are compared with those from the normal initial and rehabilitation designs, accomplished separately. Finally, the multi-objective dynamic design of the network is performed using both objective functions.

Based on the preceding discussion, six different mathematical optimization models are employed in this chapter (initial design using the first objective function, rehabilitation using the first and second objective functions, dynamic design using the first and second objective functions (each separately), and multi-objective dynamic design using both objective functions). The network hydraulic analysis is performed using the EPANET 2.0 model [59].

4 Mathematical Statement of the Optimization Problem

Equation (2) is the first objective function, the minimization of the cost of the WDN installation and rehabilitation:

$$\text{Minimize NetCost} = \text{TotalCost} - \text{CostE} + \sum_{t=1}^{T}\sum_{j=1}^{\text{Nj}} PFH_{j,t} + \sum_{t=1}^{T} PFN_t + PFT \tag{2}$$

in which NetCost = total network cost; TotalCost = total cost of the initial installation and rehabilitation; CostE = residual value; $PFH_{j,t}$ = penalty value for junction j in year t if its pressure is below the minimum; PFN_t = penalty value in year t for practical constraint; PFT = penalty value for practical constraint (these penalty

functions are defined in Eqs. (9)–(13); Nj = number of junctions; and T = total number of years in the design horizon.

Equations (3)–(5) are related to the economic (cost) calculations of the model and are defined as

$$\text{TotalCost} = \sum_{i=1}^{Ni} C(D_{i,1})_i + \sum_{t=\beta 1}^{T-(\beta 2+1)} \left[\frac{\sum_{i=1}^{Ni} \text{Cost}_{i,t}}{(1+ir)^{t-1}} \right] \tag{3}$$

$$\text{CostE} = \sum_{t=\beta 1}^{T-(\beta 2+1)} \left[\frac{\sum_{i=1}^{Ni} (\text{Cost}_{i,t}) \cdot ((T - t_{i,t}^{\text{op}})/T)}{(1+ir)^{t-1}} \right] \tag{4}$$

$$\text{Cost}_{i,t} = \begin{cases} C(D_{i,t}) \cdot L_i & \text{if DecVar}_{i,t} \neq \text{DoN} \\ 0 & \text{if DecVar}_{i,t} = \text{DoN} \end{cases} \tag{5}$$

where $C(D_{i,t})$ = cost of pipe i installed in year t with diameter D; L_i = length of pipe i; ir = discount rate; $t_{i,t}^{\text{op}}$ = number of operational years for the pipe that is changed with a new one in year t in link i; DecVar = decision variable; and Ni = number of pipes.

The second objective function is the maximization of WDN reliability, expressed by Eq. (6). Equations (7)–(8) define the concept of fuzzy reliability index (FRI) [33].

$$\text{Maximize Reliablity } (FRI) = \left(\sum_{t=1}^{T} \sum_{j=1}^{Nj} fri_{j,t} \right) \times \text{Min}(fri_{j,t}) - PFT - \sum_{t=1}^{T} PFN_t \tag{6}$$

in which FRI = fuzzy reliability index of network, $fri_{j,t}$ = fuzzy reliability index of junction j in year t; Min($fri_{j,t}$) = the lowest among all $fri_{j,t}$ for all junctions and all years. The sum and minimum values of the $fri_{j,t}$ for all nodes of the network during its operation period are considered in Eq. (6). The maximization of reliability in Eq. (6) maximizes the total fuzzy reliability index of each node in all years, and, in addition, the minimum value of $fri_{j,t}$ (which may have a severe impact on the total reliability of the system) is taken into consideration.

The membership value $\text{MemF}_{j,t}$ (obtained from Eq. 8) is multiplied by two coefficients to yield $fri_{j,t}$ as shown in Eq. (7). The first coefficient is a demand coefficient that increases the importance of nodes with larger values of demand while the second one is a coefficient introduced by Prasad and Park [56] which guarantees having reliable loops in the network:

$$fri_{j,t} = \mathrm{MemF}_{j,t} \times \left(1 - \frac{q^*_{j,t}}{\sum_{j=1}^{Nj} q^*_{j,t}}\right) \times \left(\frac{\sum_{r=1}^{\mathrm{NP}_{j,t}} \mathrm{DD}_{rj,t}}{\mathrm{NP}_{j,t} \times \mathrm{DDMax}_{j,t}}\right) \tag{7}$$

$$\mathrm{MemF}_{j,t} = \begin{cases} 0 & \text{If } h_{j,t} \le h^{\mathrm{low}}_{j,t} \\ \left(\frac{0.01}{h^*_{j,t} - h^{\mathrm{low}}_{j,t}}\right) \times \left(h_{j,t} - h^{\mathrm{low}}_{j,t}\right) & \text{If } h^{\mathrm{low}}_{j,t} < h_{j,t} \le h^*_{j,t} \\ 0.01 + \frac{1.98}{h^{**}_{j,t} - h^*_{j,t}}\left(h_{j,t} - h^*_{j,t}\right) & \text{If } h^*_{j,t} < h_{j,t} \le \frac{h^*_{j,t} + h^{**}_{j,t}}{2} \\ 0.01 + \frac{1.98}{h^*_{j,t} - h^{**}_{j,t}}\left(h_{j,t} - h^{**}_{j,t}\right) & \text{If } \frac{h^*_{j,t} + h^{**}_{j,t}}{2} < h_{j,t} \le h^{**}_{j,t} \\ \left(\frac{0.01}{h^{**}_{j,t} - h^{\mathrm{high}}_{j,t}}\right) \times \left(h_{j,t} - h^{\mathrm{high}}_{j,t}\right) & \text{If } h^{**}_{j,t} < h_{j,t} \le h^{\mathrm{high}}_{j,t} \\ 0 & \text{If } h_{j,t} \le h^{\mathrm{high}}_{j,t} \end{cases} \tag{8}$$

in which $\mathrm{MemF}_{j,t}$ = fuzzy membership function of junction j in year t; $q^*_{j,t}$ = nodal demand value for junction j in year t; $\mathrm{DD}_{rj,t}$ = diameter of pipe r connected to junction j in year t; $\mathrm{NP}_{j,t}$ = number of pipes connected to junction j in year t; $\mathrm{DDMax}_{j,t}$ = maximum diameter of pipes connected to junction j in year t; $h^*_{j,t}$ and $h^{**}_{j,t}$ = minimum and maximum required pressures in junction j in year t, respectively; $h_{j,t}$ = supplied pressure in junction j in year t; and $h^{\mathrm{low}}_{j,t}$ and $h^{\mathrm{high}}_{j,t}$ = very low and very high allowable pressures, respectively.

As remarked earlier, the hydraulic reliability expressed by the *FRI* index is identified using fuzzy logic. The fuzzy membership function is stated in Eq. (8) and shown in Fig. 1. The best membership value belongs to the junction with pressure equal to the average of the minimum and maximum allowable pressures as shown in Fig. 1 and Eq. (8). In this way, the pressures of the network junctions are placed midway between the minimum and maximum ranges, and, therefore there will be a low probability of supplying the consumers with low pressure or damaging the network because of having excessive pressure.

Equations (9)–(13) are constraints of the optimization model.

$$\text{If } \mathrm{DecVar}_{i,t} \ne \mathrm{Don}, \text{ then } \quad \mathrm{DecVar}_{i,t+(1,2,\ldots,\beta 4)} = \mathrm{DoN} \tag{9}$$

$$PFH_{j,t} = \begin{cases} \alpha_1 \left(h^*_{j,t} - h_{j,t}\right)^{\alpha_2} & \text{if } h_{j,t} < h^*_{j,t} \\ 0 & \text{if } h_{j,t} \ge h^*_{j,t} \end{cases} \tag{10}$$

$$PFN_t = \begin{cases} \alpha_3 (\mathrm{Nch}_t - \beta_3 \times Ni)^{\alpha_4} & \text{if } \mathrm{Nch}_t > \beta_3 \times Ni \\ 0 & \text{if } \mathrm{Nch}_t > \beta_3 \times Ni \end{cases} \tag{11}$$

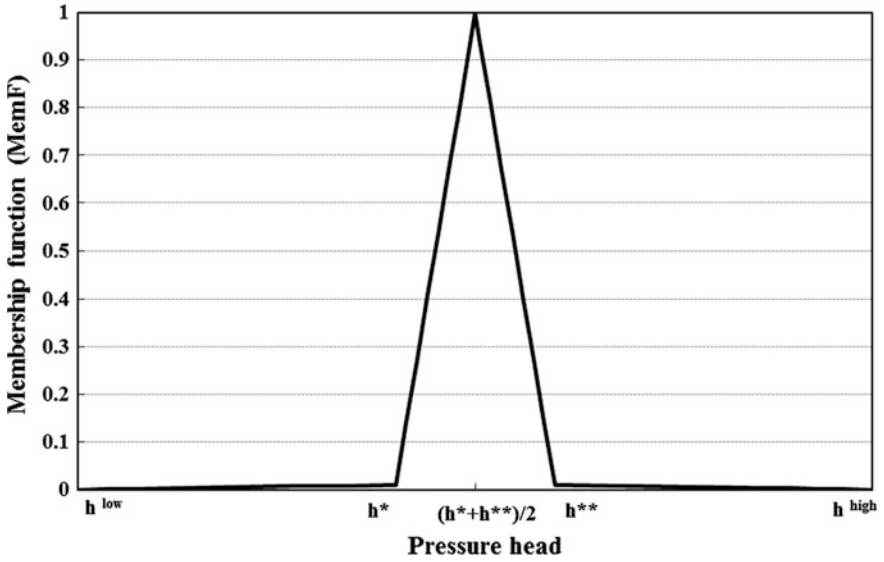

Fig. 1 Values of the membership function related to fuzzy reliability index (*FRI*)

$$PFT = \begin{cases} \alpha_5(\mathrm{TNch} - \beta_5)^{\alpha_6} & \text{if TNch} > \beta_5 \\ 0 & \text{if TNch} \leq \beta_5 \end{cases} \tag{12}$$

$$\mathrm{TNch} = \sum_{t=1}^{T} \mathrm{Nch}_t \tag{13}$$

where Nch_t = number of changes (implemented rehabilitation alternatives) in year t; TNch = total number of changes; and α_k = constant coefficient to determine the effect of the penalty function on the value of the objective function (without dimension). Equation (14) expresses head loss functions:

$$hf_{i,t} = \omega L C_{i,t}^{\mathrm{HW}-1.852} D_{i,t}^{-4.871} Q_{i,t}^{1.852} \quad \forall i = 1,2,\ldots,Ni \tag{14}$$

in which $Q_{i,t}$ = flow discharge in pipe i in year t; $hf_{i,t}$ = head loss of pipe i in year t; $C_{i,t}^{\mathrm{HW}}$ = Hazen–Williams coefficient in pipe i in year t; and ω = constant coefficient introduced by Savic and Walters [61] to be used in the Hazen–Williams equation.

Equation (15) is the geometric pattern for demand growth based on geometric population growth developed by Seifollahi-Aghmiuni et al. in 2011 [63] and Eq. (16) is the pattern for the decrease of the Hazen–Williams coefficient reported by Sharp and Walski in 1988 [66].

$$q_{j,t}^{*} = \mathrm{Exp}\left(\mathrm{Ln}(q_{j,T}^{*}) - K_g \times (T - t)\right) \tag{15}$$

in which K_g = geometric growth rate coefficient.

$$C_{i,t}^{\mathrm{HW}} = 18.0 - 37.2 \log_{10}\left(\frac{e0_i + a_i t}{D_{i,t}}\right) \tag{16}$$

where $e0_i$ = initial roughness in pipe i at the time of installation (when it was new); a_i = roughness growth rate in pipe i; and t = the number of years elapsed from initial installation ($t < T$).

5 Single-/Multi-objective HBMO Algorithm

A single-objective version of the HBMO algorithm was introduced by Bozorg Haddad et al. in 2006 [10]. The HBMO algorithm is one of the metaheuristic algorithms inspired from the honeybees life and reproduction. The algorithm starts with three user-defined parameters and one predefined parameter. The predefined parameter is the number of workers, representing the number of heuristics encoded in the program. The three user-defined parameters are the number of queens, the queen's spermatheca size representing the maximum number of matings per queen in a single mating-flight, and the number of broods that will be born by all queens. The energy and speed of each queen at the start of each mating-flight is initialized at random. A set of queens is then initialized at random. Then a randomly selected heuristic is used to improve the genotype of each queen, assuming that a queen is usually a good bee. A number of mating-flights are then undertaken. In each mating-flight, all queens fly based on their energies and speeds, where both energy and speed are generated at random for each queen before each mating-flight commences. At the start of a mating-flight, a drone is generated at random and the queen is positioned over that drone. The transition made by the queen in space is based on her speed. Therefore, at the start of a mating-flight, the speed is usually high and the queen makes moves through long steps. While the energy of the queen decreases, the speed decreases and, as a result, the neighborhood covered by the queen decreases. At each step made by the queen in space, the queen mates with the drone encountered at that step using the probabilistic rule in Eq. (17), shown below. If the mating is successful (i.e., the drone passes the probabilistic decision rule), the drone's sperm is stored within the queen's spermatheca. We may notice here that each time a drone is generated, half of his genes are marked at random since each drone is haploid by definition. Therefore, the genes that are transmitted to the broods are fixed for each drone.

The probability of mating between each drone with the queen is defined by the following equation:

$$\text{prob}(Q, D) = e^{\frac{-\Delta f}{s(t)}} \tag{17}$$

where $\text{prob}(Q, D)$ = probability of mating between drone D and queen Q, or the probability of a successful mating; Δf = value of the difference between fitness value of the drone ($f(D)$) and fitness value of the queen ($f(Q)$); $S(t)$ = queen's speed at time t. At the beginning of the mating-flight when the queen's speed is high or when the drone is fit enough, the probability of mating is high. After each moving of the queen in space or after each mating, its energy and speed decrease according to the following equations:

$$E(t+1) = E(t) - \gamma \tag{18}$$

$$S(t+1) = \alpha \times S(t) \tag{19}$$

where α is a coefficient between (0,1) and β is the value of the energy decrease.

When all queens complete their mating-flight, they start breeding. For a required number of broods, a queen is selected in proportion to her fitness and inseminated with a randomly selected sperm from her spermatheca. A worker is chosen in proportion to its fitness to improve the resultant brood. After all broods have been generated, they are sorted according to their fitness. The best brood replaces the worst queen until there is no brood that is better than any of the queens. Remaining broods are then killed and a new mating-flight starts until all assigned mating-flights are completed or convergence criteria met [9, 10]. In recent years, several investigators have implemented the HBMO algorithm to solve different problems and its good performance has been established earlier (e.g., [1, 2, 7, 9–13, 15, 16, 38, 69]).

MOHBMO, which is the multi-objective version of the HBMO algorithm, is inspired from the natural life of honeybees. Each bee hive can contain one or more queens. In the mating season, queens exit the hive and perform the mating-flight by attracting drones. The interesting point is that successful drones which were able to mate with the queen do not necessarily belong to the queen's hive. In other words, the queen might have mated with drones belonging to other hives. In addition, two queens might not live in the same hive. It means that if a hive contains two queens, one of them leaves the hive and migrates to another bee hive which does not contain a queen. It is possible for the migratory queen to mate also with drones of a new hive, leading to the replacement of honeybee genes between different bee hives. This natural process is used to develop the multi-objective version of the HBMO algorithm.

In MOHBMO, at first two hives with two (or more) different queens are generated. Each hive performs a single-objective optimization with one objective function and tries to improve its queen after a predefined number of iterations. In a multi-objective problem, the final Pareto solutions found by the algorithm will not expand more than the area between the solutions found by solving the single-objective problem with each objective function. Thus, to have a well-distributed and expanded Pareto it is

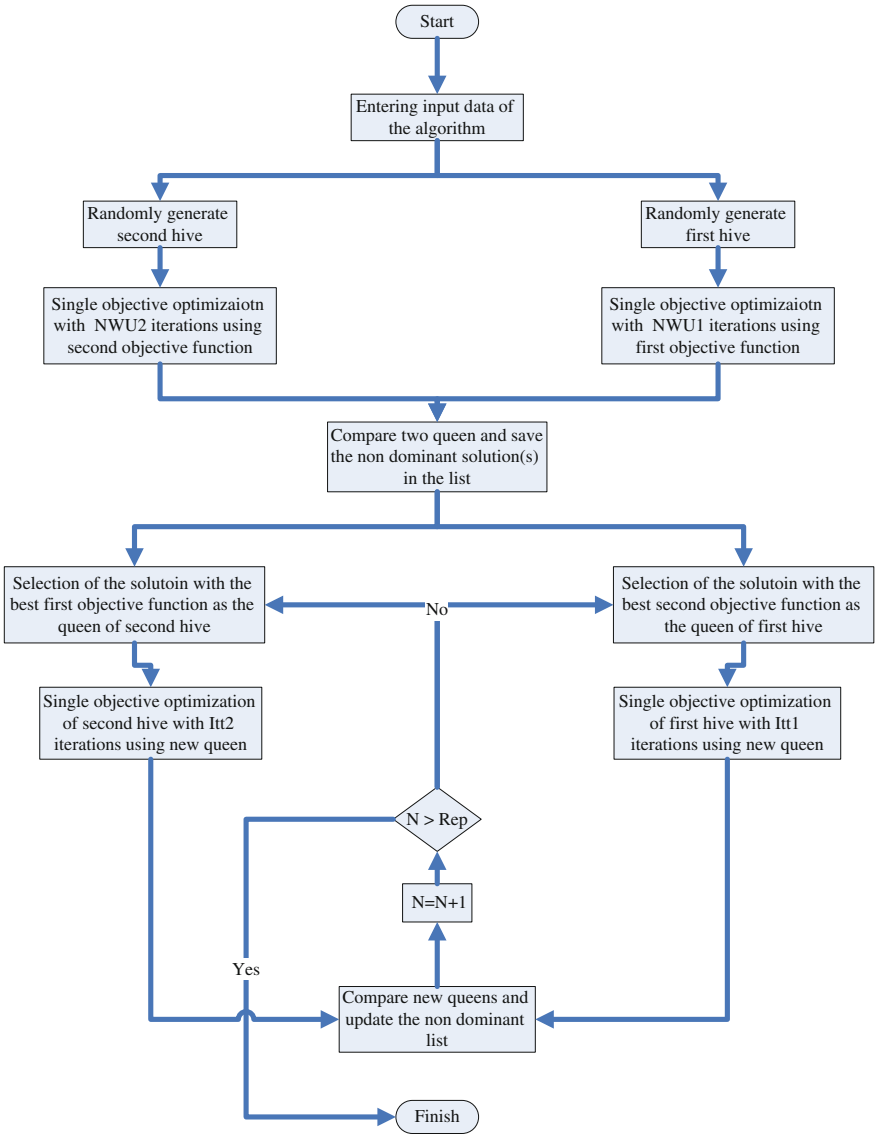

Fig. 2 Flowchart of the multi-objective honeybee mating optimization algorithm

necessary for two heads (maximum and minimum edges) of the Pareto to get as close as possible to the solution found by the single-objective solution of the problem. To ensure the aforementioned Pareto condition, a warm up (WU) period is added to the MOHBMO algorithm [25]. During the WU period, each hive tries to improve the queen through the global optimum of its own objective function, by performing a considerable number of iterations. The result is in the form of two near-optimal

solutions, one for each of the two objective functions. These solutions are compared with each other. If one of them dominates the other, it is then added in the non-dominated list; otherwise both are added to the list. The queens are then replaced in the hives. It means that the solution which is better in the first objective function will move to the hive which improves the second objective function, and vice versa. Again, each hive starts evolution through its own objective function using a new queen, while at this stage the number of iterations is much less than the WU period. After conducting a predefined number of iterations, the final solutions are again compared and moved to the nondominant list. Each new member which is added to the nondominant list will be compared with older ones and finally only the non-dominant solutions remain in the list.

After updating the nondominant list, the solution having a better value of the first objective function moves into the second hive (that improves the second objective function), and vice versa. Once more, each hive evolves to improve its own objective function and this repetitive procedure continues until a stopping criterion, which is usually a predefined number of iterations, is reached. If the WU period is successfully performed and near-optimum single-objective results are obtained, and if the number of each hive's iteration during the repetitive procedure and stopping criterion are chosen properly, one can expect to obtain a well-distributed and expanded Pareto solution. Figure 2 shows a flowchart of the MOHBMO algorithm, in which NWU1 and NWU2 are the number of iterations in WU periods for the first and second hives, respectively. Also, Itt1 and Itt2 define the number of each hive's iterations during the repetitive procedure, and Rep is the number of repetitions of the algorithm (stopping criterion).

6 DSS Application

The DSS in this chapter contains a: (1) Policy named dynamic design method with a new fuzzy reliability index (*FRI*); and (2) DSS tool which is the combination of a single-/multi-objective optimization algorithm and a hydraulic simulator (EPANET 2.0) supported initially by long-term data such as nodal demands (predicted using Eq. 15) and pipes' roughness (predicted using Eq. 16) during the useful life of each network, and any other needed data. The DSS is applied to two sample networks named Two-Loop and Hanoi water distribution networks. The results are as follows.

6.1 The Two-Loop Network

An example of a two-loop network [3] is now considered. The network (Fig. 3) consists of eight pipes and seven nodes (junctions). One of the nodes is the reservoir and other ones are demand nodes. The data for this benchmark problem can be found in Alperovits and Shamir [3].

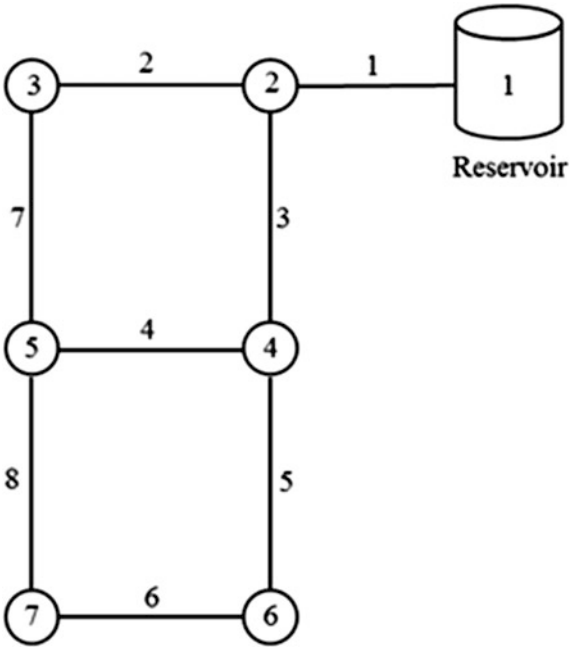

Fig. 3 Schematic of the two-loop network

The minimum and maximum acceptable pressures are 30 and 60 m above ground level, respectively. There are 14 commercially available diameters for this network and each diameter cost has been reported by Alperovits and Shamir [3]. These costs apply to purchasing pipes and installing them. This enables the comparison of the results of dynamic and normal designs. The discount rate (*ir*) is 0.05 [10]. The design horizon (T) equals 30 years while the K_g coefficient (see Eq. 15) is equal to 0.1. Figure 4 shows the increasing trend of nodal demands for the two-loop network during its service life using Eq. (15).

The initial roughness ($e0$) and roughness growth rates (a) are found in Bozorg Haddad et al. [10]. The Hazen–Williams coefficient for all pipes equals 130 in the 1st year [3] and it changes thereafter according to Eq. (16) as shown in Fig. 5. Finally, the maximum number of rehabilitation activities implemented during the operation period (β_5) is assumed to be equal 10.

Using the demand data of the final operation period (year) and a Hazen–Williams coefficient equal to 130 the design of the two-loop network was performed by the normal design method with the objective function of minimizing costs. The result obtained by the HBMO was \$419,000, which has been reported by other investigators (e.g., [20, 24, 32, 61, 73]).

Figure 6 shows the change in nodal pressure heads in the designed network during its operation period.

It is seen in Fig. 6 that the nodal pressures in the 1st years of operation exceed 30 m. In the following years pressure heads are near 30 m and decrease in the final

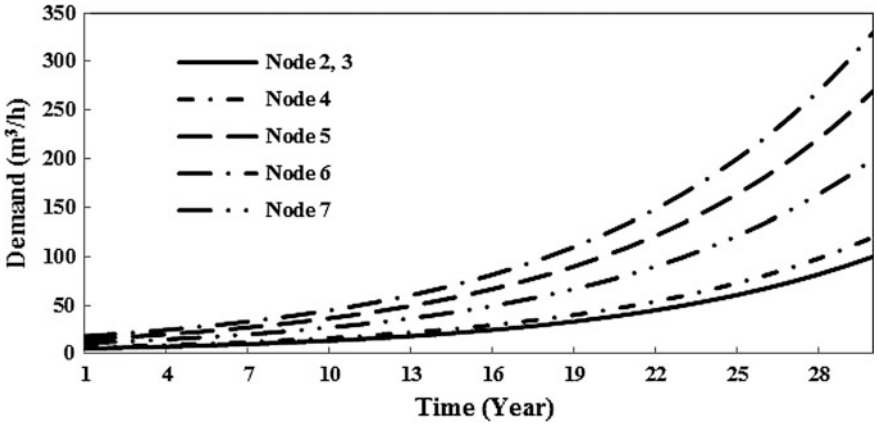

Fig. 4 Changes of nodal demands in the two-loop network during the operational period

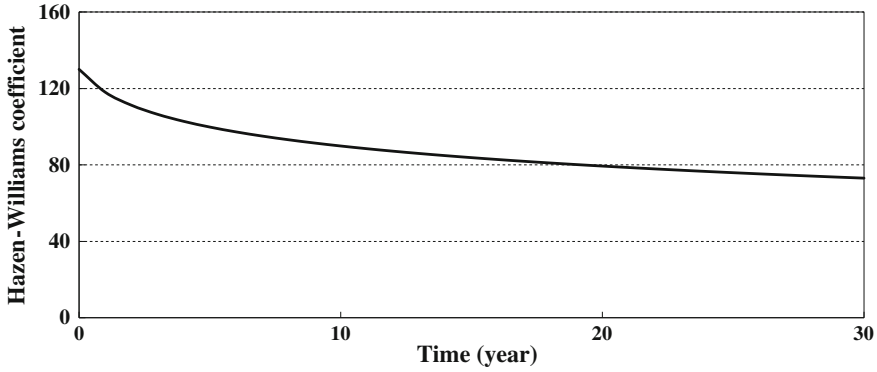

Fig. 5 Change of the Hazen–Williams coefficient in the two-loop network during operational period

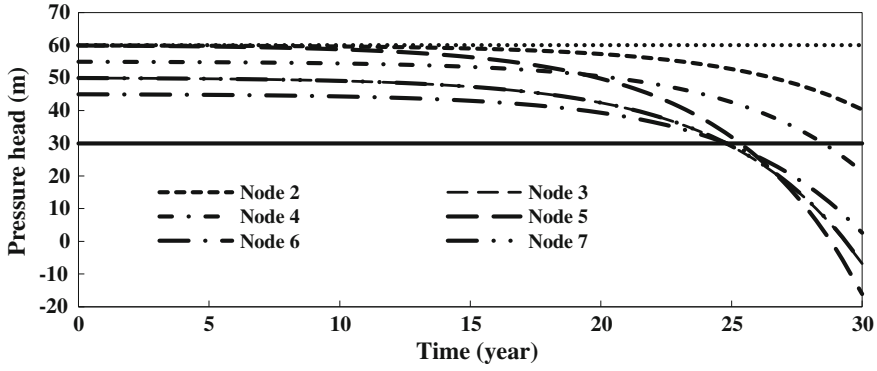

Fig. 6 Changes of nodal pressure heads in the two-loop network from initial design method without rehabilitation during operational period

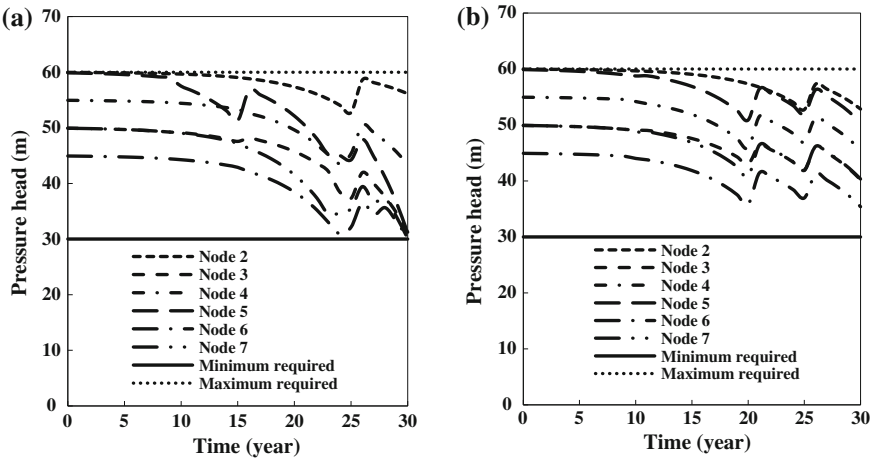

Fig. 7 Nodal pressure heads in the two-loop network for normal design during the operational period, **a** minimizing the total network cost; **b** maximizing reliability

years. The rise in pressure head is caused by increased nodal water demands and pipe roughness. The rise in pressure heads requires rehabilitation intervention. Rehabilitation was performed using the HBMO algorithm with two objective functions, each optimized separately. The results are shown in Fig. 7. This figure shows the condition of the designed network, after rehabilitation with each objective function. It is seen in Fig. 7 that after rehabilitation the pressure values are placed in the allowable range during the operational period. In addition, rehabilitating the network with the aim of maximizing the fuzzy reliability index leads to adequate nodal pressures. The post-rehabilitation pressure heads are close to the average of the minimum and maximum pressure heads. The values of the *FRI* for the designs considering the first and second objective functions are 0.58 and 0.76, respectively.

The dynamic design of this network was also performed with the two objective functions, each optimized separately. The HBMO algorithm was implemented with both objective functions, each optimized separately. Figure 8 shows the pressure conditions of the results obtained from HBMO with each objective function. Clearly, the pressure values fall in the allowable range with either objective function. The satisfactory order and arrangement of the nodal pressures in the dynamic design with the second objective function is noticeable, also. A comparison of Figs. 7b and 8b shows that the dynamic design method is able to deliver nodal pressures that are closer to the average desirable value. A comparison of Figs. 7b and 8b shows the superiority of the dynamic design over the normal design from the viewpoint of the reliability level.

FRI values for the solutions found with dynamic and normal designs with the two objective functions are also listed in Table 1. The values in Table 1 establish

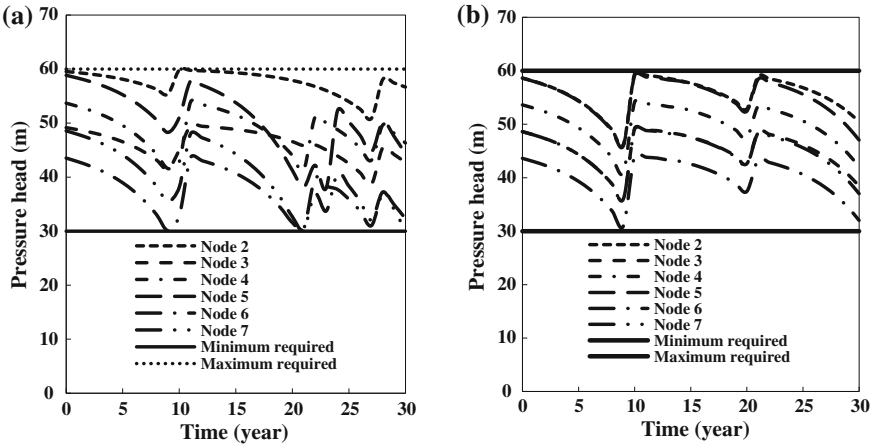

Fig. 8 Nodal pressures in the two-loop network from dynamic design during the operational period, **a** minimizing the total network cost; **b** maximizing reliability

Table 1 Value of the *FRI* from dynamic design and normal design with both objective functions in the two-loop network

Objective function	Dynamic design	Normal design
Minimize cost	0.65	0.58
Maximize *FRI*	4.9	0.76

that the *FRI* values from both design scenarios (first and second objective functions) of the dynamic design are superior to those from the normal design.

Figure 9 shows the graph of expenditures for the initial design and rehabilitation activities of the two-loop network from the normal and dynamic designs using the cost minimization objective function. This graphical comparison clearly shows the difference between normal and dynamic designs. It is seen that the initial expenditure from the dynamic design is approximately half of the initial cost from that associated with the normal design. On the other hand, the rehabilitation costs from the dynamic design are larger than those corresponding to the normal design. Thus, the dynamic design of the network is able to postpone the project's expenditure and consequently it is possible to decrease the total network cost of the project due to the influence of the discount rate. Tables 2 and 3 show the diameters associated with the initially installed and rehabilitated pipes from the dynamic design of the two-loop network. During the rehabilitation period ($10 \leq t \leq 28$ years), numbers labeled by a star are the pipe diameters which were replaced while numbers defined by two stars are those which were added (duplicated).

A fact worthy of notice in Table 3 is the value of pipe diameters in year zero. Pipe 1 has a diameter equal to 0.20 m (8 in.) while distal pipes have greater pipe diameters such as 0.61 m (24 in.), which is not a common design. This is caused by the use of the *FRI* as the objective function in such a small network. The best value

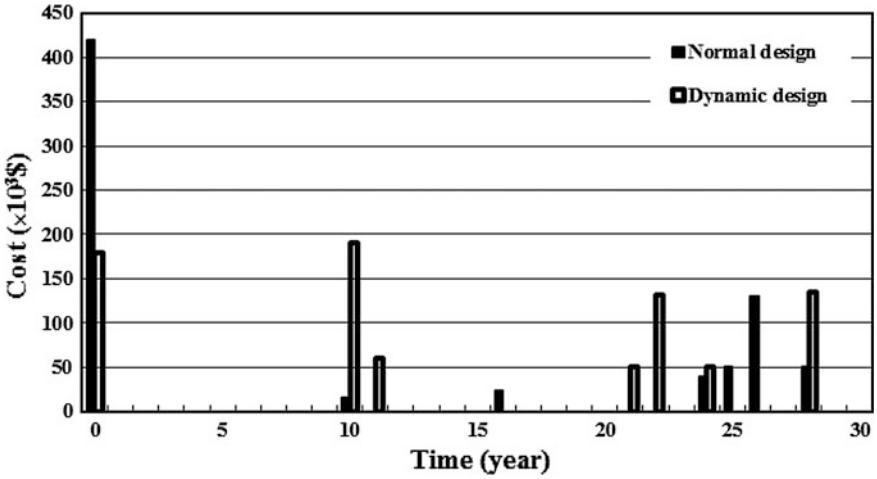

Fig. 9 Expenditures for the initial design and rehabilitation activities in the two-loop network from the normal and dynamic designs using the cost minimization objective function

for nodal pressure is the mean of the minimum and maximum permissible pressure heads in the definition of the FRI index. As a result, using the *FRI* as the objective function of the optimization model, the pipe diameters are chosen in a way to produce mean nodal pressure values for all junctions. Because in this small example, the whole network is highly influenced by the first pipe; this pipe has been chosen to have the minimum diameter and maximum head loss to control the downstream nodal pressures. However, this may not happen in more realistic networks, especially those having pump stations and valves (see next case study). These results highlight the necessity of multi-objective dynamic design, which is presented later on.

A comparison of expenditures is listed in Table 4, showing that the initial cost from the dynamic design is less than that from the normal design while its rehabilitation cost is larger. The NetCost represents the sum of initial and rehabilitation costs minus the residual value in Table 4. NetCost from the two-loop network's dynamic design is approximately 24 % less than its normal design. If the residual value is not included, NetCost from the dynamic design is approximately 37 % less than NetCost of normal design.

6.2 *The Hanoi Network*

The second problem is the WDN of Hanoi city, Vietnam which is shown in Fig. 10 [31]. This network contains 32 nodes that consists one reservoir and 34 pipes organized in 3 loops. No pumping facilities are considered for the network since only a single fixed-head reservoir at elevation of 100 m is available. The minimum

Table 2 Pipe diameters obtained from dynamic design with the objective of cost minimization in the two-loop network (in mm)

Year	Pipe							
	1	2	3	4	5	6	7	8
0	254.0	254.0	25.4	152.0	102.0	254.0	254.0	203.0
1	–	–	–	–	–	–	–	–
2	–	–	–	–	–	–	–	–
3	–	–	–	–	–	–	–	–
4	–	–	–	–	–	–	–	–
5	–	–	–	–	–	–	–	–
6	–	–	–	–	–	–	–	–
7	–	–	–	–	–	–	–	–
8	–	–	–	–	–	–	–	–
9	–	–	–	–	–	–	–	–
10	457.0[a]	–	–	–	356.0[a]	–	–	–
11	–	–	305.0[b]	–	–	–	102.0[a]	–
12	–	–	–	–	–	–	–	–
13	–	–	–	–	–	–	–	–
14	–	–	–	–	–	–	–	–
15	–	–	–	–	–	–	–	–
16	–	–	–	–	–	–	–	–
17	–	–	–	–	–	–	–	–
18	–	–	–	–	–	–	–	–
19	–	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–	–
21	–	–	–	–	305.0[b]	–	–	–
22	–	–	457.0[b]	–	–	–	25.4[a]	–
23	–	–	–	–	–	–	–	–
24	–	–	–	305.0[b]	–	–	–	–
25	–	–	–	–	–	–	–	–
26	–	–	–	–	–	–	–	–
27	–	–	–	–	–	–	–	–
28	457.0[b]	–	–	–	–	–	–	51.0[b]
29	–	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–	–

[a]Replaced pipes
[b]Added (duplicated) pipes

and maximum permissible pressure heads are 30 and 60 m for each node, respectively ($h_j^* = 30$ and $h_j^{**} = 60$). The Hazen–Williams coefficient value equals 130. Other relevant data are listed in Fujiwara and Khang [31]. The value of *ir*, e_0, α (roughness growth rate in pipes of network), and k_g were chosen in the same

Table 3 Pipe diameters obtained from dynamic design with the objective of *FRI* maximization for the two-loop network (in mm)

Year	Pipe							
	1	2	3	4	5	6	7	8
0	203.2	355.6	355.6	609.6	609.6	609.6	457.2	609.6
1	–	–	–	–	–	–	–	–
2	–	–	–	–	–	–	–	–
3	–	–	–	–	–	–	–	–
4	–	–	–	–	–	–	–	–
5	–	–	–	–	–	–	–	–
6	–	–	–	–	–	–	–	–
7	–	–	–	–	–	–	–	–
8	–	–	–	–	–	–	–	–
9	–	–	–	–	–	–	–	–
10	355.6[a]	–	–	609.6[b]	–	–	–	–
11	–	–	–	–	609.6[b]	–	–	609.6[b]
12	–	457.2[a]	–	–	–	–	–	–
13	–	–	–	–	–	–	–	–
14	–	–	508.0[a]	–	–	–	–	–
15	–	–	–	–	–	–	–	–
16	–	–	–	–	–	–	–	–
17	–	–	–	–	–	–	–	–
18	–	–	–	–	–	–	–	–
19	–	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–	–
21	406.4[b]	–	–	609.6[b]	–	–	–	–
22	–	–	–	–	609.6[b]	–	–	609.6[b]
23	–	–	–	–	–	–	–	–
24	–	–	–	–	–	–	–	–
25	–	–	–	–	–	–	–	–
26	–	–	–	–	–	–	–	–
27	–	–	–	–	–	–	–	–
28	–	–	–	–	–	–	–	–
29	–	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–	–

[a]Replaced pipes
[b]Added (duplicated) pipes

manner as done with the two-loop network. The value of β_5 was assumed to be equal to 32.

The Hanoi network was optimized with normal design to obtain its minimum cost diameter set. The relevant network cost of this design equaled \$6,000,448, which has been the least cost reported in the literature using different optimization

Table 4 Optimization results from dynamic and normal designs with the objective of cost minimization in the two-loop network

	Initial design ($)	Rehabilitation cost ($)	Scrap value ($)	Net cost ($)
Normal design	419,000	96,533	69,661	445,872
Dynamic design	180,000	265,321	104,637	340,685

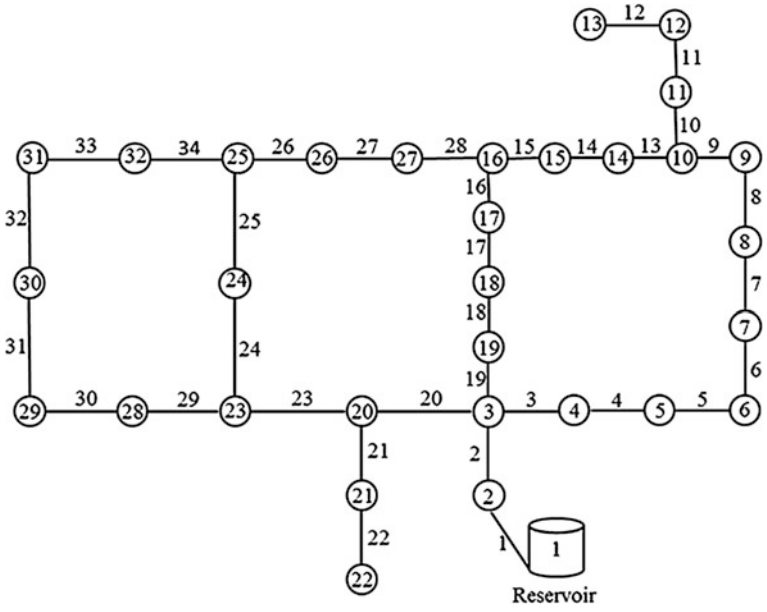

Fig. 10 Schematic of the Hanoi network

algorithms [34]. By considering changes in nodal demands and roughness coefficient (as done with the two-loop network), the calculated change of some of the network's nodal pressures is shown in Fig. 11. Only the boundary nodes of the 32-node network are graphed in Fig. 11. It is seen in Fig. 11 that the performance of this normally designed network will decrease considerably in the final years of operation. The nodal pressure heads in the initial years are about 90 m because of the topographic and hydraulic conditions prevailing in the Hanoi network. Pressure-reducing valves (PRV) and pumping stations are required to reduce the high pressure heads.

The Hanoi network was also designed using the dynamic design concept. This was accomplished with both objective functions, each optimized separately. Cost and *FRI* values of the dynamic designed network using each objective function are shown in Table 5. It is seen in Table 5 that the cost of a dynamically designed network (during 30 years of operation) is less than the normal design of the Hanoi

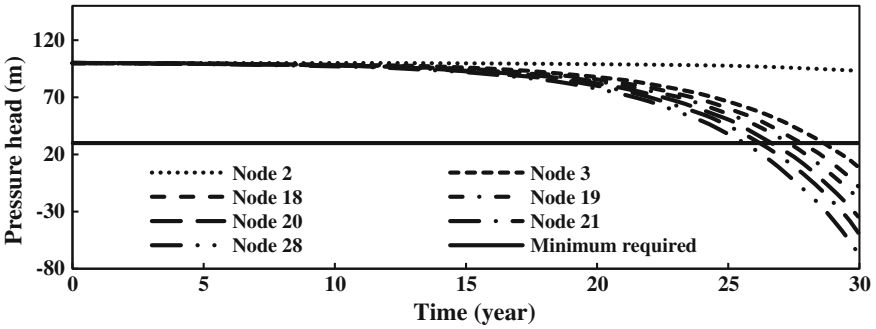

Fig. 11 Changes of nodal pressure heads in Hanoi network from normal design without rehabilitation during operational period

Table 5 Cost and *FRI* values from the dynamically designed Hanoi network optimizing each objective function separately

Objective function	Cost ($)	*FRI*
FRI maximization	10,670,531	1.36
Cost minimization	4,910,814	0.40

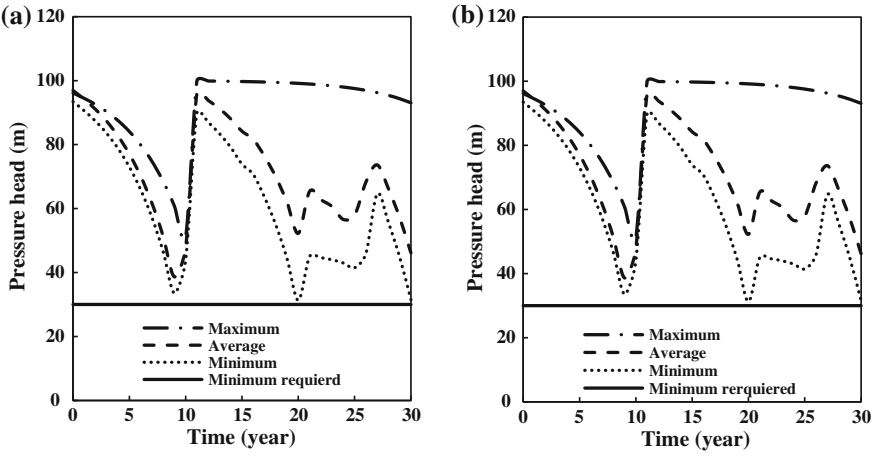

Fig. 12 Envelope curve of nodal pressure in the Hanoi network from dynamic design during the operational period, **a** minimizing the total network cost; **b** maximizing reliability

network. On the other hand, by designing the network using the *FRI* objective function, a reliability index value was obtained equal to 1.36 with a cost equal to $10,670,531.

Figure 12 depicts envelope curves of the nodal pressures of the dynamically designed Hanoi network optimizing each objective function separately. By comparing Fig. 12 with Fig. 11, it is noticeable that the nodal pressures of the network

during its useful life are kept at an acceptable level above 30 m. It is worth mentioning that the performance of the system during the remaining years of operation can be improved by solving the model every few years, using new data. In this way, the model incorporates field conditions as they unfold over time.

6.3 Multi-objective Approach

The single-objective normal and dynamic designs of networks were presented and compared for two case studies in previous sections. Yet, the nature of this problem is multi-objective because the two objectives (cost minimization and reliability maximization) give rise to trade-offs between them. Therefore, using a multi-objective approach it is possible to determine those trade-offs. In this section the results of multi-objective design of the two-loop and Hanoi networks are presented using the multi-objective honeybee mating optimization algorithm (MOHBMO). The objectives are Eq. (2) (cost minimization) and Eq. (6) (maximization of reliability). Constraints and relationships are the same as single-objective problems. The multi-objective solutions with normal design and dynamic design can then be compared to assess their relative virtues or limitations. First, however, we compare the Pareto space of normal and dynamic design methods.

The first step is to make an initial multi-objective optimal design of the two-loop network using the two proposed objective functions. The initial design of the network is performed by considering the nodal demands at the end of the WDN's service life (nodal demands reported by Alperovits and Shamir [3]) and the results are the network's optimal pipe diameters. Ten runs of the MOHBMO algorithm were conducted to initially design the two-loop network in which the algorithm simultaneously attempts to optimize the objective functions of cost minimization and *FRI* maximization. The results of these 10 different runs produced the Pareto frontier for this problem shown in Fig. 13. It is worth mentioning that all the answers of this Pareto are nondominant and also feasible solutions.

The two extreme Pareto points in Fig. 13 (points A and B) were obtained using the single-objective design optimization models for the two-loop network, i.e., optimizing each objective function separately. Point A represents the network designed in previous sections using the cost minimization objective function and simulated/rehabilitated during the operational period (the results of simulation and rehabilitation were shown in Figs. 6 and 7). The other extreme Pareto, point B, represents a network designed using the *FRI* maximization objective function. Clearly, as the reliability of the system increases, its cost increases too. The decision maker must choose a combination of cost and reliability found on the Pareto frontier.

In this study, the conflict resolution method proposed by Young in 1993 [79] is used to choose a point among different solutions of Pareto. According to the conflict resolution method, a desirability function is added to each objective function. By

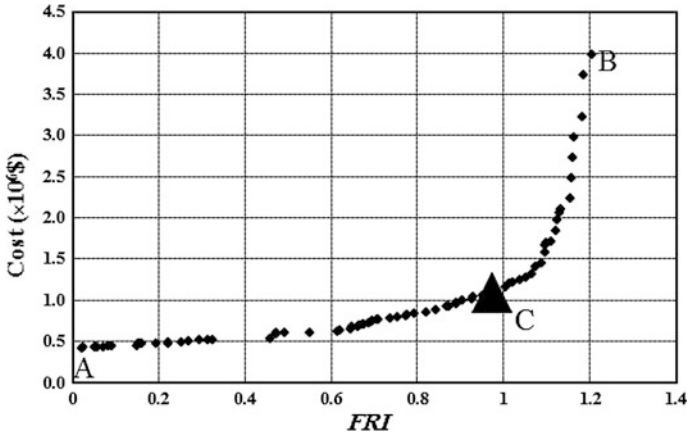

Fig. 13 Pareto frontier of cost and reliability index in the two-loop network corresponding to the initial design

Table 6 Pipe diameters, network cost, and *FRI* for solutions A, C, and B: two-loop network

Point	Diameter (mm)		
Pipe no.	A	C	B
1	457.2	457.2	457.2
2	254.0	508.0	609.6
3	406.4	508.0	609.6
4	101.6	355.6	609.6
5	406.4	457.2	609.6
6	254.0	457.2	609.6
7	254.0	508.0	609.6
8	25.4	457.2	609.6
Cost ($)	419,000	1,090,000	3,980,000
FRI	0.02	0.97	1.2

maximizing an equation based on the gradient of different points of the Pareto, the best point is then chosen [79]. Using this method, point C, which is labeled by a triangle in Fig. 13, is chosen. Naming A as the solution determined by the objective function of cost minimization, B as the solution achieved by *FRI* maximization, and C as the chosen answer using Young's method [79], pipe diameters, network's total costs, and *FRI* values are shown in Table 6.

Those three designs differ with respect to pipe diameters and system reliability. Therefore, their performances during the operational and rehabilitation period are different from the cost and *FRI* reliability index viewpoints. Solution A has the lowest cost and also the worst *FRI* index value. As a result, this solution is expected to need considerable rehabilitation during the operational period. In contrast, network B is more expensive but has better reliability than other Pareto solutions.

Table 7 Optimization results of normal (solutions A, C, and B) and dynamic designs showing the values of the two objective functions

Design type	Initial cost of network installation in year zero ($)	Rehabilitation cost ($)	Residual value ($)	Net cost of network installation and rehabilitation (Normal design) or dynamic design ($)
B	3,980,000	278,522	1,424,801	2,833,721
C	1,090,000	144,918	315,934	918,984
A	419,000	96,533	69,661	445,872
Dynamic design	180,000	265,321	104,637	340,685

Therefore, this solution B requires less rehabilitation while its total cost is higher than the cost of other Pareto solutions. Clearly, solution C is a compromise between cost and reliability.

The comparison of the Pareto spaces of the normal and dynamic design methods is performed using all three benchmark solutions (A, B, and C in Table 6) as different conditions for the normal design. That is, each of three solutions is separately simulated during 30 years of the operational period and they are rehabilitated if any sign of failure or underperformance appears. Results of these three normal designs are then compared with those of the dynamic designed network.

Solutions C and B were simulated during the operational period (same as solution A in Sect. 6.1) to determine the necessity of rehabilitation for each design. The condition and changing manner of annual nodal pressures in all three solutions show the necessity of rehabilitation as previously shown in Fig. 6 for the least-cost initial design of the two-loop network (point A). Nodal pressures of the design points C and B, the same as point A, decrease considerably during the operational years, and, therefore, all three solutions involve rehabilitation.

Rehabilitation design for solutions C and B are performed using the single-objective HBMO algorithm with the two objective functions, each optimized separately, as was previously done for solution A. For each case, 10 different runs were conducted and the best results are considered. The normal design (initially designed and then rehabilitated) of all three networks was performed using both objective functions. Table 7 presents a comparison of results of the normal (solutions A, C, and B) and dynamic designs of the two-loop network (the dynamic designs were described in Sect. 2), using the minimum cost objective function. It is now possible to compare the Pareto space of the normal design with that of the dynamic design.

Figure 14 shows results of the optimization problem of the two-loop network design for normal (solutions A, C, and B) and dynamic designs. Each pair of points shown in Fig. 14 is relevant to the single-objective normal/dynamic design of the two-loop network optimizing each objective function separately.

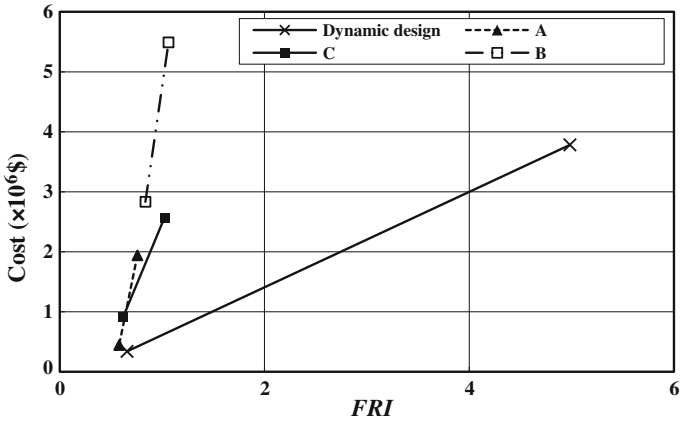

Fig. 14 Results of the single-objective optimization problem in the two-loop network design from normal and dynamic designs

It is seen in Fig. 14 that the dynamic design was able to improve the cost and *FRI* in comparison with the normal design (solutions A, B, and C). In other words, a single-objective design of the two-loop network to minimize the network's cost using the dynamic design method yielded the least cost in comparison with solutions A, C, and B. Similarly, dynamic design reached a considerably better fuzzy reliability index (*FRI*) than normal designs. Figure 14 indicates that if the two-loop network is designed in a multi-objective manner, the Pareto set obtained from the dynamic design method will dominate other Pareto sets yielded by normal designs. In other words, the Pareto frontier from the dynamic design will dominate all other points existing on other Paretos.

The comparison presented in Fig. 14 also shows the influence of the initial design on the design of rehabilitation period. It is seen in Fig. 14 that the normally designed network is worse than the dynamic design judged by economics and hydraulic reliability, in all three initially designed networks using cost minimization (solution A), *FRI* maximization (solution B), and trade-off between these two objectives (solution C). As a result, it is concluded that economic and hydraulic conditions of the network during its total useful life will definitely improve if the network is initially designed by considering rehabilitation during its service life.

The third prominent feature observed in Fig. 14 is the comparison between the broadness of the dynamic and normal-design Paretos. Figure 14 establishes that the dynamic design Pareto is broader than the Paretos obtained from normal designs. That is, the dynamic design provides a wider range of solutions to decision makers. The close proximity of normal-design Pareto solutions (pair of nodes for solutions A, C, and B) in comparison with dynamic design, renders the multi-objective design of networks A, C, and B meaningless. Thus, in this study the MOHBMO algorithm is useful for the multi-objective dynamic design of the two-loop network, only.

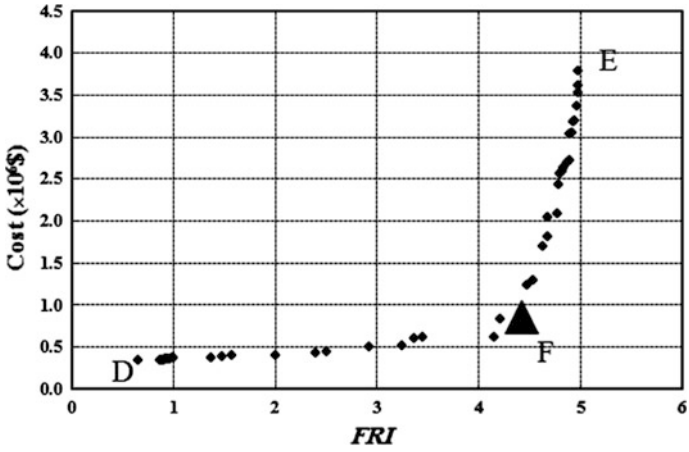

Fig. 15 Pareto frontier of cost and reliability index in the two-loop network from dynamic design

To perform the multi-objective dynamic design of the two-loop network, 10 different runs were conducted using the MOHBMO algorithm with the objective functions of cost minimization and *FRI* maximization. The 10 Pareto solutions are shown in Fig. 15. All the points on this Pareto frontier are feasible solutions. The two extremes of the Pareto frontier presented in Fig. 15 (points D and E) are the same points shown for the dynamic design in Fig. 14, obtained by optimizing each objective function separately. The methodology of Young in 1993 [79] is again used to select one of the solutions of the Pareto set of Fig. 15. The selected point (point F) is labeled with a triangle in this figure. This solution costs $850,000 and its *FRI* value is 4.43. Hence, by spending about 20 % of the cost of solution E, solution F can supply 80 % of its reliability. In contrast, the network cost relevant to solution F is about 2.5 times the cost of solution D while its *FRI* value is 7 times greater than the *FRI* value of solution D.

The multi-objective dynamic design of the Hanoi network was performed using the the MOHBMO algorithm. Once more, 10 different runs were conducted and the combination of these 10 runs is portrayed in Fig. 16. The two extremes of this Pareto frontier are also the solutions obtained from the single-objective design of the Hanoi network optimizing each objective function separately. The cost of solution H in this Pareto is equal to $4,910,814 while its *FRI* value is about 0.4. The values for the cost and *FRI* of solution I are $10,670,531 and 1.36, respectively.

Young's method [79] was used to choose a solution from the Hanoi network Pareto frontier. The selected point is shown by a triangle in Fig. 16. Solution J, with about 60 % of the cost of solution I, can supply 80 % of its reliability. In contrast, the network cost relevant to solution J is about 1.5 times of the cost of solution H while its *FRI* value is 1.7 times higher than the *FRI* value of solution H.

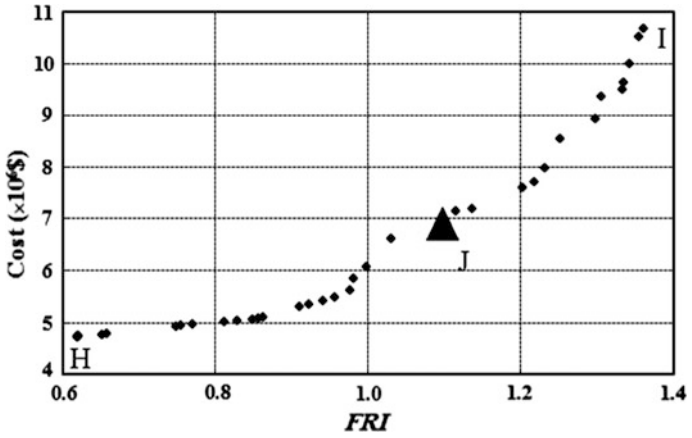

Fig. 16 Pareto frontier of cost and reliability index for the Hanoi network from dynamic design

7 Conclusion

This chapter presented a DSS for the long-term design and operation of water distribution networks demonstrating the importance and usefulness of the simultaneous optimization of initial design and rehabilitation of WDNs. Changes in a network's condition during its operational period were simulated and initially introduced to the DSS using equations suggested in previous studies. The simultaneous initial and rehabilitation design of two case studies was performed as the main DSS planning strategy by proposing a method named dynamic design of WDNs. The WDN optimization problem was posed as a single- and multi-objective optimization problem with minimum cost and maximum fuzzy reliability index as two separate objectives. The HBMO and MOHBMO algorithms were used as the optimization tools for solving this discrete, nonlinear, complex, and large-scale design and operation problem. Although an all-encompassing, quantitative DSS is yet to be developed, the approach described in this chapter takes into account the structural and hydraulic state of the network while providing a framework for the future inclusion of other considerations as well. Results indicated that the dynamic design method is able to decrease the total capital cost of the WDN installation and rehabilitation while it can improve the performance of the system from the viewpoint of reliability. The simultaneous initial and rehabilitation designs of the network made it possible to postpone the installation cost by considering the network's condition in future years, leading to considerable savings in the project. Additionally, multi-objective dynamic design of networks produces larger solution spaces than single-objective optimization.

The DSS based on the dynamic design method encompasses three separate fields of WDN studies, namely, initial network design, rehabilitation activities, and prediction of different operational period tasks such as hydraulic conditions or

water-demand volumes. Each of these research fields is being separately studied by investigators around the world. The dynamic design method is likely to evolve as new findings emerge from new research.

References

1. Afshar, A., Bozorg Haddad, O., Marino, M.A., Adams, B.J.: Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation. J. Franklin Inst. **344**(5), 452–462 (2007)
2. Afshar, A., Shafii, M., Bozorg Haddad, O.: Optimizing multi-reservoir operation rules: an improved HBMO approach. J. Hydroinformatics **13**(1), 121–139 (2010)
3. Alperovits, E., Shamir, U.: Design of optimal water distribution systems. Water Resour. Res. **13**(6), 885–900 (1977)
4. Alvisi, S., Franchini, M.: Multiobjective optimization of rehabilitation and leakage detection scheduling in water distribution systems. J. Water Res. Plann. Manage. **135**(6), 426–439 (2009)
5. Arulraj, G.P., Suresh, H.R.: Concept of significance index for maintenance and design of pipe networks. J. Hydraul. Eng. (ASCE) **121**(11), 833–837 (1995)
6. Berardi, L., Kapelan, Z., Giustolisi, O., Savic, D.A.: Development of pipe deterioration models for water distribution systems using EPR. J. Hydroinformatics **10**(2), 113–126 (2008)
7. Bozorg Haddad, O., Marino, M.A.: Dynamic penalty function as a strategy in solving water resources combinatorial optimization problems with honey-bee optimization (HBMO) algorithm. J. Hydroinformatics **9**(3), 233–250 (2007)
8. Bozorg Haddad, O., Mariño, M.A.: Optimum operation of wells in coastal aquifers. Proc. Inst. Civil Eng. Water Manage. **164**(3), 135–146 (2011)
9. Bozorg Haddad, O., Afshar, A., Mariño, M.A.: Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization. Water Resour. Manage. **20**(5), 661–680 (2006)
10. Bozorg Haddad, O., Adams, B.J., Mariño, M.A.: Optimum rehabilitation strategy of water distribution systems using the HBMO algorithm. J. Water Supply Res. Technol. **151**, 337–350 (2008)
11. Bozorg Haddad, O., Afshar, A., Marino, M.A.: Honey-bee mating optimization (HBMO) algorithm in deriving optimal operation rules for reservoirs. J. Hydroinformatics **10**(3), 257–264 (2008)
12. Bozorg Haddad, O., Afshar, A., Marino, M.A.: Design-operation of multi-hydropower reservoirs: HBMO approach. Water Resour. Manage. **22**(12), 1709–1722 (2008)
13. Bozorg Haddad, O., Afshar, A., Marino, M.A.: Optimization of non-convex water resource problems by honey-bee mating optimization (HBMO) algorithm. Eng. Comput. (Swansea, Wales), **26**(3), 267–280 (2009)
14. Bozorg Haddad, O., Moradi-Jalal, M., Mirmomeni, M., Kholghi, M.K.H., Mariño, M.A.: Optimal cultivation rules in multi-crop irrigation areas. Irr. Drain. **58**(1), 38–49 (2009)
15. Bozorg Haddad, O., Mirmomeni, M., Mariño, M.A.: Optimal design of stepped spillways using the HBMO algorithm. Civil Eng. Environ. Syst. **27**(1), 81–94 (2010)
16. Bozorg Haddad, O., Mirmomeni, M., ZarezadehMehrizi, M., Mariño, M.A.: Finding the shortest path with honey-bee mating optimization algorithm in project management problems with constrained/unconstrained resources. Comput. Optim. Appl. **47**(1), 97–128 (2010)
17. Bozorg Haddad, O., Moradi-Jalal, M., Mariño, M.A.: Design-operation optimisation of run-of-river power plants. Proc. Inst. Civil Eng. Water Manage. **164**(9), 463–475 (2011)
18. Clark, R.M., Stafford, C.L., Goodrich, J.A.: Water distribution systems: a spatial and cost evaluation. J. Water Resour. Plann. Manage. (ASCE) **108**, 243–256 (1982)

19. Cunha, M.C., Sousa, J.: Hydraulic infrastructures design using simulated annealing. J. Infrastruct. Syst. (ASCE) **7**(1), 32–39 (2001)
20. Cunha, M.C., Sousa, J.: Water distribution network design optimization: simulated annealing approach. J. Water Resour. Plann. Manage. (ASCE) **125**(4), 215–221 (1999)
21. Dandy, G.C., Engelhardt, M.: Multi-objective trade-offs between cost and reliability in the replacement of water mains. J. Water Resour. Plann. Manage. ASCE **132**(2), 79–88 (2006)
22. Elstad, J.C., Byer, P.H., Adams, B.J.: Optimal timing of the restoration of watermain carrying capacity. In: Proceedings of the CSCE Centennial Conference, Montreal, P.Q., pp. 60–70 (1987)
23. Engelhardt, M.O., Skipworth, P.J., Savic, D.A., Saul, A.J., Walters, G.A.: Rehabilitation strategies for water distribution networks: a literature review with a UK perspective. Urban Water **2**, 153–170 (2000)
24. Eusuff, M.M., Lansey, K.E.: Optimization of water distribution network design using the shuffled frog leaping algorithm. J. Water Resour. Plann. Manage. (ASCE) **129**(3), 210–225 (2003)
25. Fallah-Mehdipour, E., Bozorg Haddad, O., Beygi, S., Mariño, M.A.: Effect of utility function curvature of Young's bargaining method on the design of WDNs. Water Resour. Manage. **25** (9), 2197–2218 (2011)
26. Fallah-Mehdipour, E., Bozorg Haddad, O., Mariño, M.A.: MOPSO algorithm and its application in multipurpose multireservoir operations. J. Hydroinformatics **13**(4), 794–811 (2011)
27. Fallah-Mehdipour, E., Bozorg Haddad, O., Mariño, M.A.: Real-time operation of reservoir system by genetic programming. Water Resour. Manage. **26**(14), 4091–4103 (2012)
28. Fallah-Mehdipour, E., Bozorg Haddad, O., RezapourTabari, M.M., Mariño, M.A.: Extraction of decision alternatives in construction management projects: application and adaptation of NSGA-II and MOPSO. Expert Syst. Appl. **39**(3), 2794–2803 (2012)
29. Fallah-Mehdipour, E., Bozorg Haddad, O., Mariño, M.A.: Developing reservoir operational decision rule by genetic programming. J. Hydroinformatics **15**(1), 103–119 (2013)
30. Fallah-Mehdipour, E., Bozorg Haddad, O., Mariño, M.A.: Extraction of multicrop planning rules in a reservoir system: Application of evolutionary algorithms. J. Irri. Drain. Eng. **139**(6), 490–498 (2013)
31. Fujiwara, O., Kang, D.B.: A two-phase decomposition method for optimal design of looped water distribution networks. Water Resour. Res. **26**(4), 539–549 (1990)
32. Geem, Z.W.: Optimal cost design of water distribution networks using harmony search. J. Eng. Optim. **38**(3), 259–280 (2005)
33. Ghajarnia, N.: Multi-objective dynamic design of water distribution networks. M.Sc. thesis, Department of Irrigation and Reclamation Engineering, University of Tehran, Iran (2009)
34. Ghajarnia, N., Bozorg Haddad, O., Mariño, M.A.: Performance of a novel hybrid algorithm in the design of water networks. Water Manage. **164**(4), 173–191 (2011)
35. Goulter, I.C., Bouchart, F.: Reliability constrained pipe networks model. J. Hydraul. Eng. (ASCE) **16**(2), 221–229 (1990)
36. Goulter, I.C., Lussier, B.M., Morgan, D.R.: Implications of head loss path choice in the optimization of water distribution networks. Water Resour. Res. **22**(5), 819–822 (1986)
37. Halhal, D., Walters, G.A., Savic, D.A., Ouazar, D.: Scheduling of water distribution system rehabilitation using structured messy genetic algorithms. Evol. Comput. **7**(3), 311–329 (1999)
38. Jahanshahi, G., Bozorg Haddad, O.: Honey-bee mating optimization (HBMO) algorithm for optimal design of water distribution systems. World Environmental and Water Resources Congress, Honolulu, Hawaii, United States, 12–16 May 2008
39. Kanakoudis, V.K.: A troubleshooting manual for handling operational problems in water pipe networks. Water Supply Res. Technol. AQUA IWAp **53**(2), 109–124 (2004)
40. Kanakoudis, V.K.: Vulnerability based management of water resources systems. Hydroinformatics WAp **6**(2), 133–156 (2004)
41. Kanakoudis, V.K., Tolikas, D.K.: The role of leaks and breaks in water networks—technical and economical solutions. Water Supply Res. Technol. AQUA IWAp **50**(5), 301–311 (2001)

42. Kanakoudis, V.K., Tolikas, D.K.: Assessing the Performance Level of a water system. Water Air Soil Pollut. **4–5**, 307–318 (2004)
43. Karimi-Hosseini, A., Bozorg Haddad, O., Mariño, M.A.: Site selection of raingauges using entropy methodologies. Proc. Instit. Civil Eng. Water Manage. **164**(7), 321–333 (2011)
44. Keen, P.G.W., Morton, M.S.S.: Decision Support Systems: An Organizational Perspective. Addison-Wesley Pub (1978)
45. Kessler, A., Shamir, U.: Analysis of the linear programming gradient method for optimal design of water supply networks. Water Resour. Res. **25**(7), 1469–1480 (1989)
46. Kim, J.H., Mays, L.W.: Optimal rehabilitation model for water-distribution systems. J. Water Res. Plann. Manage. (ASCE) **120**(5), 674–692 (1994)
47. Kleiner, Y., Adams, B.J., Rogers, J.S.: Long-term planning methodology for water distribution system rehabilitation. Water Resour. Res. **34**(8), 2039–2051 (1998)
48. Kleiner, Y., Adams, B.J., Rogers, J.S.: Selection and scheduling of rehabilitation alternatives for water distribution systems. Water Resour. Res. **34**(8), 2053–2061 (1998)
49. Lansey, K.E., Basnet, C., Mays, L.W., Woodburn, J.: Optimal maintenance scheduling for water distribution systems. Civil Eng. Syst. **9**(3), 211–226 (1992)
50. Lippai, I., Heaney, J.P., Laguna, M.: Robust water system design with commercial intelligent search optimizers. J. Comput. Civil Eng. **13**(3), 135–143 (1999)
51. Maier, H.R., Simpson, A.R., Zencchin, A.C., Foong, W.K., Phang, K.Y., Seah, H.Y., Tan, C. L.: Ant colony optimization for design of water distribution systems. J. Water Resour. Plann. Manage. **129**(3), 200–209 (2003)
52. Male, J.W., Walski, T.M., Slutski, A.H.: Analyzing watermain replacement policies. J. Water Resour. Plann. Manage. (ASCE) **116**(3), 363–374 (1990)
53. Nafi, A., Kleiner, Y.: Scheduling renewal of water pipes while considering adjacency of infrastructure works and economies of scale. J. Resour. Plann. Manage. (ASCE) **136**(5), 519–530 (2010)
54. Nafi, A., Werey, C., Llerena, P.: Water pipe renewal using a multi-objective optimization approach. Can. J. Civ. Eng. **35**, 87–94 (2008)
55. Orouji, H., Bozorg Haddad, O., Fallah-Mehdipour, E., Mariño, M.A.: Estimation of Muskingum parameter by meta-heuristic algorithms. Proc. Inst. Civil Eng Water Manage. **166**(6), 315–324 (2013)
56. Prasad, T.D., Park, N.S.: Multiobjective genetic algorithms for design of water distribution networks. J. Water Resour. Plann. Manage. (ASCE) **130**(1), 73–82 (2004)
57. Quindry, G.E., Brill, E.D., Liebman, J.C.: Optimization of looped water distribution systems. J. Environ. Eng. **107**(4), 665–679 (1981)
58. Roshani, E., Filion, Y.R.: Event based network rehabilitation planning and asset management. In: 14th annual Water Distribution Systems Analysis Conference (WDSA), Adelaide, Australia, pp. 933–943 (2012)
59. Rossman, L.A.: EPANET user manual. Drinking Water Research Division, Risk Reduction Engineering Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, Cincinnati, OH, July 2000
60. Sabbaghpour, S., Naghashzadehgan, M., Javaherdeh, K., Bozorg Haddad, O.: HBMO algorithm for calibrating water distribution network of Langarud city. Water Sci. Technol. **65** (9), 1564–1569 (2012)
61. Savic, D.A., Walters, G.A.: Genetic algorithms for least-cost design of water distribution networks. J. Water Resour. Plann. Manage. (ASCE) **123**(2), 67–77 (1997)
62. Schneiter, C.R., Haims, Y.Y., Li, D., Lambert, J.H.: Capacity reliability of water distribution networks and optimum rehabilitation decision making. Water Resour. Res. **32**(7), 2271–2278 (1996)
63. Seifollahi-Aghmiuni, S., Bozorg Haddad, O., Omid, M.H., Mariño, M.A.: Long-term efficiency of water networks with demand uncertainty. Proc. Inst. Civil Eng. Water Manage. **164**(3), 147–159 (2011)

64. Seifollahi-Aghmiuni, S., Bozorg Haddad, O., Omid, M.H., Mariño, M.A.: Effects of pipe roughness uncertainty on water distribution network performance during its operational period. Water Resour. Manage **27**(5), 1581–1599 (2013)
65. Shamir, U., Howard, C.D.D.: An analytic approach to scheduling pipe replacement. J. Am. Water Works Assoc. **71**(5), 248–258 (1979)
66. Sharp, W.W., Walski, T.M.: Predicting internal roughness in water mains. J. Am. Water Works Assoc. **80**, 34–40 (1988)
67. Shokri, A., Bozorg Haddad, O., Mariño, M.A.: Algorithm for increasing the speed of evolutionary optimization and its accuracy in multi-objective problems. Water Resour. Manage. **27**(7), 2231–2249 (2013)
68. Sol, H.G., Takkenberg C.A.Th., De VriesRobbe, P.F.: Expert systems and artificial intelligence in decision support systems. In: Proceedings of the Second Mini Euroconference, Springer, Lunteren, The Netherlands, 17–20 Nov 1985 (1987)
69. Solgi, M., Bozorg Haddad, O., Seifollahi-Aghmiuni, S., Loaiciga, H.A.: Intermittent operation of water distribution networks considering equanimity and justice principles. J. Pipeline Syst. Eng. Pract. (2015). doi:10.1061/(ASCE)PS.1949-1204.0000198
70. Soltanjalili, M., Bozorg Haddad, O., Mariño, M.A.: Operating water distribution networks during water shortage conditions using hedging and intermittent water supply concepts. J. Water Resour. Plann. Manage. (2013) In Press
71. Soltanjalili, M., Bozorg Haddad, O., Seifollahi-Aghmiuni, S., Mariño, M.A.: Water distribution networks simulation by optimization approaches. Water Sci. Technol. Water Supply (2013) In Press
72. Su, Y.C., Mays, L.W.: New methodology for determining the optimal rehabilitation and replacement of water distribution system components. In: Abt S.R., Gessler J. (eds.) Proceedings of Hydraulic Engineering ASCE National Conference, Edited by , (ASCE), New York, N.Y., 1149–1154 (1988)
73. Suribabu, C.R., Neelakantan, T.R.: Design of water distribution networks using particle swarm optimization. J. Urban Water **3**(2), 111–120 (2006)
74. Todini, E.: Looped water distribution networks design using a resilience index based heuristic approach. J. Urban Water **2**(3), 115–122 (2000)
75. Tsitsifli, S., Kanakoudis, V., Bakouros, L.: Pipe networks risk assessment based on survival analysis. Water Resour. Manage. **25**14, 3729–3746 (2011)
76. Walski, T.M.: The wrong paradigm-Why water distribution optimization doesn't work. J. Water Resour. Plann. Manage. (ASCE) **127**(4), 203–205 (2001)
77. Walski, T.M., Pelliccia, A.: Economic analysis of watermain breaks. J. Am. Water Works Assoc. **74**(3), 140–147 (1982)
78. Walski, T.M.: Optimization and pipe-sizing decisions. J. Water Resour. Plann. Manage. (ASCE) **121**(4), 340–343 (1995)
79. Young, H.P.: An evolutionary model of bargaining. J. Econ. Theory **59**, 145–168 (1993)

Application of the Simulated Annealing Algorithm for Transport Infrastructure Planning

Ana Laura Costa, Maria Conceição Cunha, Paulo A.L.F. Coelho and Herbert H. Einstein

Abstract Decisions in planning for transport infrastructure are the result of complex technical, political, and societal concerns. Its context of limited public funding and large costs require that decision making is soundly supported. When addressing real-world problems, however, it is extremely difficult to ascertain the system configuration yielding the most value. Different alternatives exist that trade-off interrelated factors governing the value of the configurations. Metaheuristics can be of assistance when solving such real-world problems. This chapter presents an application of the simulated annealing algorithm to solve an integrated approach to high-speed rail planning. The algorithm capabilities in addressing the intricacies imposed by large and complex problems are discussed.

Keywords Metaheuristics · Simulated annealing · Optimization · Parameter calibration · High-speed rail modeling

A.L. Costa · M.C. Cunha (✉) · P.A.L.F. Coelho
Department of Civil Engineering, University of Coimbra, Coimbra, Portugal
e-mail: mccunha@dec.uc.pt

A.L. Costa
e-mail: alcosta@dec.uc.pt

P.A.L.F. Coelho
e-mail: pac@dec.uc.pt

H.H. Einstein
Department of Civil and Environmental Engineering,
Massachusetts Institute of Technology, Cambridge, USA
e-mail: einstein@mit.edu

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*,
Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_11

1 Introduction

1.1 Planning for Transport Infrastructure

Building and operating large transport infrastructure is expensive. Significant investments are necessary and have to compete for the allocation of limited government resources. The context of public funding drives an increased demand for addressing key social, economic, and environmental concerns. In the case of road and railway infrastructure, the planning stage considers the appraisal of multiple infrastructure configurations and technical solutions. Interrelations between local characteristics and the infrastructure specifics can be established for determining the project solutions yielding the most value. Studying different alternatives is of paramount importance as the subsequent project design stage further details the characteristics of the infrastructure within boundaries imposed by planning.

1.1.1 The Case of High-Speed Rail: Definition of Stations

Ridership and demand capture from other modes are decisive for HSR projects. Amongst others, they are affected by the location of the HSR stations defining the connections established by the system [1]. In addition to journey-related factors such as connection time or comfort, other elements affect the decision to travel by HSR. Accessibility to HSR stations is a major factor influencing the choice of rail instead of other modes [1–3] and increasing the number of stations within the HSR system increases accessibility. On the other hand, a rising number of intermediate stations increase journey times for passengers, decreasing the attractiveness of the HSR and having a negative impact on ridership [1, 4].

These counteracting effects for the definition of which cities to connect need to be addressed at the planning stage [5], along with the political decisions that ultimately enable the HSR project [6].

1.1.2 The Case of High-Speed Rail: Layout and Technical Solutions

The connection between HSR stations is established by trains circulating, as the designation suggests, at significantly higher speeds than those operated in conventional railways. Upgraded existing lines support speeds of circa 200 km/h while specially built lines are generally equipped for speeds in excess of 250 km/h [7].

The high operating speeds require that the HSR infrastructure complies with strict layout specifics, particularly for the radii of horizontal curves and the longitudinal gradient of linear sections. Large centrifugal acceleration imposed on trains, as a result of high speeds and small radii of horizontal curves, compromises the quality of the service and, in extreme cases, safety. It can also increase track degradation and reduce the components lifetime. The radii of horizontal curves

should be as large as possible: standards define comfort values that can be exceptionally reduced due to project difficulties until an absolute minimum radii imposed by safety concerns [8]. Conversely, smaller longitudinal gradients desirable and absolute maximum gradients are established [7]. It is advantageous, however, to implement minimum gradients that facilitate drainage. Steeper slopes result in increased energy consumption and braking distances for downward trains that may reduce the line capacity, particularly for freight and passenger-mixed traffic lines.

In addition to the strict requirements imposed on the HSR layout, further difficulties are imposed by the deployment site. Large variations of ground elevations, irregular land-use patterns, heterogeneous and hazardous geotechnical conditions, and population density are common challenges to HSR projects. These are overcome by technical solutions including embankments, cuts, retaining walls, bridges, and tunnels. In each case, however, strict displacement limits of the HSR tracks need to be assured [9] through adequate correspondence between the technical solution and the deployment site characteristics. The multiple layout possibilities for connecting any two HSR stations coupled with the interrelations between infrastructure and local characteristics make the HSR planning process a complex one.

1.1.3 The Case of High-Speed Rail: Cost Overview

A comprehensive model to estimate the overall costs of intercity transportation systems is proposed by Levinson et al. [10] considering social costs in addition to construction, operation, and maintenance costs. Applications to HSR by Levinson et al. in 1997 [11] suggest the HSR fares well in social costs, including pollution, accidents, and noise compared with air travel and highways. Nonetheless, protests against HSR configuration have occurred [12]. In Europe, United States, and China the concerns stated include alteration of landscapes and proximity to schools, issues for which alternative solutions can be studied at the planning stage.

Conversely, the large HSR infrastructure capital cost is the main disadvantage compared to air travel and highway systems. Such costs, however, can vary greatly. Based on data from worldwide HSR projects, Campos and de Rus in 2009 [12] observe HSR construction costs ranging from €4.7 million to €65.8 million (2005) per km. These large variations relate to intrinsic characteristics of the projects. Topographic characteristics, HSR layout, and technical solutions and the type of traffic explain major differences within Europe [12]. Compare the adoption of larger longitudinal gradients in France's HSR passenger dedicated line, avoiding the expensive major tunnel and viaduct construction, with Germany's ICE mixed traffic and stricter layout restrictions coupled with corridors over mountainous terrains [5].

1.1.4 Complexity for High-Speed Rail Decision Making

In general, the higher the operating speed is the stricter the design requirements are, resulting in large construction and maintenance costs of an HSR system. While large expenditures are required, the HSR has advantages over other transportation modes and the decisions at the planning stage, varying the intermediate cities connected, the corridors, and the technical solutions have an important influence on the resulting costs and HSR performance.

Multiple alternatives can be studied that trade-off interrelated factors governing the value of the HSR solutions. This can hardly be achieved without the support of comprehensive tools capable of integrated approaches that systematically address the complexities involved in decision making.

1.2 Optimization of Linear Transport Infrastructure

Mathematical optimization models are widely used in engineering for addressing large and complex problems. An objective function expresses the measure of wealth to be optimized by changing design variables and subject to constraints that define the feasibility of the solutions. However complex the formulated models may be, these are nonetheless representations of reality. The ability to express the real problems depends on the simplifications assumed, in this case the representation of the linear transport infrastructure problem. Once a satisfactory formulation is established for the problem, it is necessary to solve the optimization model and find the solution yielding the most value. Multiple solving techniques exist and, among others, the choice should take into consideration the ability to deal with the model specifics.

1.2.1 Representation of Problem Specifics

Several models have been developed for the optimization of linear transport infrastructure, particularly for highway alignment optimization in detailed stages of the problem.

Jong et al. in 1998 [13] propose a model for the optimization of horizontal and three-dimensional highway alignments considering the minimization of construction costs, user costs, and location-dependent costs subject to restrictions of the highway geometry layout and location. Geographic Information Systems (GIS) integration is not considered [13] and difficulties exist in representing location cost items. An important concern when modeling linear transport infrastructure is the interrelation of the infrastructure characteristics and the site conditions that are subject to irregular patterns and large variations. GIS integration is developed by Jong et al. in 2000 [14] for optimization of horizontal alignments. However, by optimizing horizontal alignments only, limitations are imposed for the highway

construction cost formulations as these depend on the highway elevation relatively to the ground. Other approaches have been developed that focus on modeling the optional connection of intermediate locations and infrastructure specifics [15, 16], environmental concerns [17, 18], or cost formulations [19–21].

At the smaller planning scale Gipps et al. in 2001 [22] develop a comprehensive framework for the optimization of linear transport infrastructures. It considers the influence of variable geotechnical behavior influencing the technical solutions and their construction and maintenance costs. This is an important factor for realistically assessing construction costs that is not considered in the preceding examples. Gipps et al. in 2001 [22], however, do not integrate the connection of intermediate locations and the infrastructure alignment definition, which are two intertwined issues, particularly for HSR planning [4].

The examples presented are not exhaustive but demonstrate limitations and capabilities of existing models. It is shown that the consideration of an integrated three-dimensional approach including discontinuous cost items is required for realistically modeling the problem. The lack of a comprehensive cost formulation leads to suboptimal solutions. However, when considering comprehensive cost formulations for the optimization of linear transport infrastructure, the cost function becomes an implicit function of the decision variables, resulting in non-differentiable and noisy cost functions [23]. These complexities need to be addressed by the solving techniques.

1.2.2 Solving Techniques

Various solving techniques have been considered. Parker in 1977 [24] proposed a method for solving highway corridor optimization combining linear programming and shortest path algorithms. The linearity requirements allow only for a limited set of problem costs and constraints to be considered and difficulties exist in dealing with backward bending alignments. The latter limitation has also been observed in solving alignment optimization problems with dynamic programming [25]. However, challenging land-use patterns or topographic conditions may impose infrastructure configurations that significantly deviate from the direct straight path between two points. Studies by Chew et al. in 1989 [26] used gradient-based search methods to solve highway alignment optimization problem. The need to assess cost gradients has limitations in addressing discontinuous cost functions that are needed to realistically formulate the problem. Kang et al. in 2012 [27] further discuss this, observing limitations in various search methods other than metaheuristics for addressing the complexity of optimization models that realistically represent the problem.

Metaheuristics including particle swarm, genetic algorithms, and simulated annealing [14, 27–30] have been successfully applied to linear transport infrastructure optimization problems. Among other characteristics, flexibility is provided by metaheuristics for taking into account the specifics of real cases. In this chapter, an application of the simulated annealing algorithm is discussed for solving a three-dimensional approach to high-speed rail planning optimization.

2 The Simulated Annealing Algorithm

2.1 Overview

The simulated annealing algorithm is inspired by the annealing process of materials from high energy states into low energy states used extensively in the steel and glass industries. In this process, the material is initially heated until reaching a high enough temperature that allows the particles to move. Slow and controlled cooling follows that then allows the particles to rearrange into configurations corresponding to very low energy states. Lower temperatures represent lower energy states but slow cooling is necessary to avoid obtaining suboptimal configurations. This concept is expressed by the Metropolis algorithm [31] that has been generalized by Kirkpatrick et al. in 1983 [32] to solve the travelling salesman problem. The applications by Kirkpatrick et al. in 1983 [32] and later Cerny in 1985 [33] to solve the travelling salesman problem draw an analogy between obtaining the lowest energy configuration of a system and achieving the global optimum solution for the optimization problem [34]. In the analogy, the solutions to the optimization problem correspond to the configurations of a physical system and the objective function value of a solution corresponds to the energy of a configuration.

According to the Metropolis algorithm, if a current system configuration i of energy E_i is rearranged into a configuration j of energy E_j, the probability p that j will be the new current system configuration in a minimization problem is given by Eq. (1).

$$p = \min\left[1, \exp\left(-\left(E_j - E_i\right)/(k_B t)\right)\right] \tag{1}$$

where k_B is the Boltzman's constant and t is the temperature.

Apart from the physical analogy, the simulated annealing algorithm can be regarded as a stochastic technique that applies a probabilistic mechanism for accepting worsening solutions [35]. As a local search technique, the simulated annealing algorithm considers new candidate configurations obtained by perturbing a current configuration within a neighborhood structure. This corresponds to small changes within the vicinity of the current configuration. Therefore, if the current configuration is suboptimal but corresponds to a local optimum, it is necessary to accept worsening configurations in order to continue the search for a global optimum. This is illustrated in Fig. 1.

Consider solving a minimization problem. By using the Metropolis algorithm, in addition to accepting transitions that improve the objective function ($E_j < E_i$), the simulated annealing also accepts worsening transitions ($E_j > E_i$). The probability of accepting worse solutions in Eq. (1) is given by $p = \exp(-(E_j - E_i)/(k_B t))$ and decreases as the algorithm progresses. This acceptance rate is governed by the decrease of a control parameter, the temperature in Eq. (1). Even if the algorithm performance is dissociated from underlying assumptions of physical meaning, terminology borrowed from the annealing physical process is used [36].

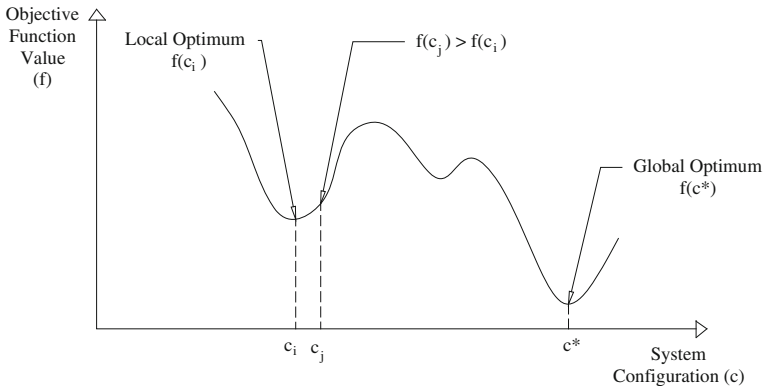

Fig. 1 Accepting worsening system configurations to escape local optima

Aarts et al. in 1997 [34] and Dekkers and Aarts in 1991 [35] discuss how, in mathematically modeling the SAA using Markov chains, for certain conditions, asymptotic convergence to the global optimum can be proven to have probability 1. A Markov chain can be defined as a sequence of trials from a sampling process in which the probability of a particular trial result depends only on the previous trial result and not on the preceding sequence according to Eq. (2) [34].

$$P_{ij}(k) = P\{X(k) = j \,|\, X(k-1) = i\} \tag{2}$$

where $P_{ij}(k)$ is the transition probability from result i to result j at the kth trial and $X(k)$ is a random variable defining the result of the kth trial.

Based on these principles, simulated annealing algorithms have been successfully implemented to solve multiple engineering problems [37]. However, Dekkers and Aarts in 1991 [35] and Johnson et al. in 1989 [36] discuss that considering an infinite number of homogeneous Markov chains (in which the transition probability does not depend on the trial number k) of infinite length, as required by for the asymptotic convergence proof, is impracticable. Thus, implementations in finite time with finite length Markov chains at a finite number of descending values of the control parameter are required. The convergence of the algorithm is governed by a set of parameters defining a cooling schedule that establishes how the control parameter, temperature, is decreased. These cannot guarantee global optimum convergence but its probability may increase according to the specifics of the implementation [35].

2.2 *Considerations for the Algorithm Implementation*

The implementation of the simulated annealing algorithm involves the definition of an initial system configuration, the neighborhood structure and the generation process of new candidate configurations, and the cooling schedule. The following

subsections discuss issues to consider when defining these elements. While the algorithm may be applied for solving multiple problems, the best annealing algorithm parameters in each case depend on the problem solved and its size [36].

2.2.1 Initial System Configuration

Heuristic methods, arbitrary definition, random or best guess approaches have been used in the literature to derive initial system configurations for implementing the simulated annealing algorithm. It may be advantageous to start the algorithm with a better than random configuration, particularly if it takes advantage of specific problem structures rather than employing a general heuristic [36]. However, heuristic defined initial configurations can add overhead time to the algorithm implementation. On the other hand, if the algorithm implementation allows the search of the global feasible search space of the problem, the initial system configuration will not limit the search nor will it influence the quality of the solutions produced [38]. The generation of the initial system configuration should thus trade-off overhead and running times, taking into account the problem specifics.

2.2.2 Generating New System Configurations

The generation of new candidate system configurations within a neighborhood structure aims at obtaining transitions that correspond to rearrangements instead of profound transformations. The neighborhood structure imposes boundaries to the extent of each modification of the current configuration. In addition, the degree of freedom, the number of changes at each transition, of the process influences the algorithm performance [39]: minor changes in the configuration transitions may limit the algorithm to explore only a limited part of the search space while the opposite profound changes may lead to a random behavior and not taking advantage of the neighborhood structure. The procedures to generate feasible candidate configurations of the system are required to address these concerns and further address difficulties that may be imposed by the problem feasibility constraints.

2.2.3 Cooling Schedule

The cooling schedule defines the evolution of the control parameter (temperature) throughout the algorithm and the number of system configurations analyzed at each temperature step. Two main classes of schedules exist [34, 36]: static cooling schedules and dynamic cooling schedules. Static cooling schedules implement fixed parameters which are not changed during the algorithm implementation. Dynamic schedules implement parameters governing the temperature decrease that are adapted as the algorithm progresses.

A general approach to the cooling schedule defines criteria for initial and final system temperatures, the rate with which the initial temperature is decreased until reaching the final temperature and a minimum number of system configurations to be evaluated at each temperature step. Figure 2 illustrates how the cooling schedule governs the inner loops performed within a generic implementation of the simulated

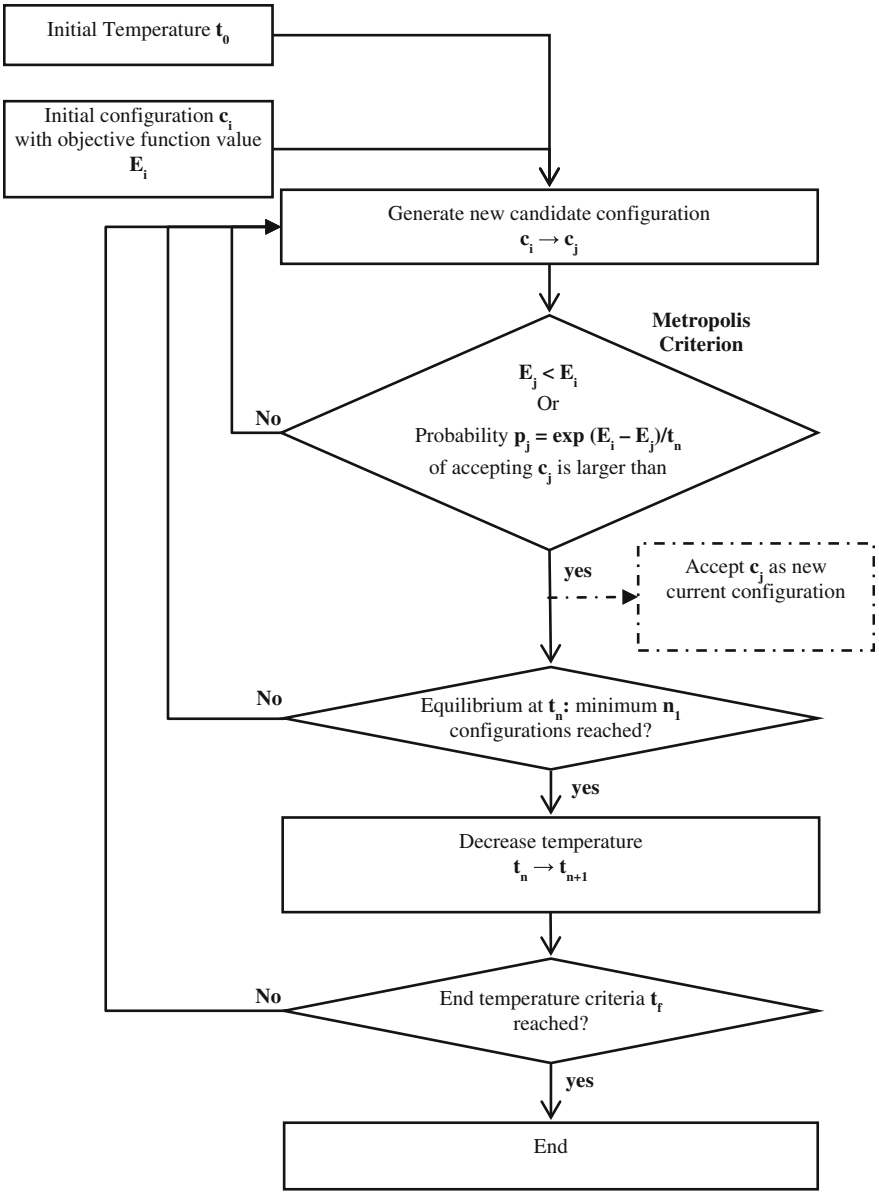

Fig. 2 Flowchart of a generic implementation of the simulated annealing algorithm

annealing algorithm. The flowchart in Fig. 2 considers a minimization problem in which worse solutions correspond to larger objective function values.

The initial temperature should be such that the system is considered, in analogy with the physical annealing process, "melted". The temperature at each step influences the probability of accepting configurations with worse objective function values (see Metropolis criterion in Fig. 2) and thus a sufficiently large temperature, for which almost all system configurations are accepted, should be chosen initially. Kirkpatrick in 1984 [40] suggested an initial temperature that allows at least 80 % of the transitions to be accepted at the initial temperature step.

A temperature decrease occurs when equilibrium is reached at a given temperature step (Fig. 2). The implementation of static geometric cooling is common [36], where temperature at the $n + 1$ step, $tn + 1$ is given by $tn + 1 = r.\ tn$ at a constant rate $0 < r < 1$. The rate with at the temperature is decreased has a substantial influence on the quality of the solutions obtained [34, 36]. If the temperature decreases too fast, the algorithm will accept only a small number of worsening configurations and the performance will be similar to iterative improvement, which terminates at the first local optimum. Conversely, if the decrease rate is too small, the algorithm performance can resemble a random search where virtually any neighboring configuration is accepted.

Different cooling schedules can be implemented, including linear or logarithmic cooling [36] or more complex dynamic schedules based on the standard deviation of the objective function values of the Markov chain [35].

The equilibrium considered in the finite-time implementation of simulated annealing relates to the number of transitions at each temperature step. A fixed number of transitions per temperature can be established or adaptive schedules with varying number of transitions can be implemented. The amount of time that the algorithm spends between very high and very low temperatures affects the quality of the solutions and adaptive cooling schedules can be tailored to such purpose [32, 36].

The termination criteria establish when to stop the algorithm. It relates to the temperature value corresponding to a "frozen" state for which no further improvements of the objective function are achieved. This has been implemented in the literature based on different criteria, including the average value of the objective function [35] and a minimum number of temperature decreases below a percentage threshold of accepted transitions [36].

3 Application for Planning Transport Infrastructure

This section discusses the implementation of the simulated annealing algorithm for a transport infrastructure planning problem. The application considers the optimization model proposed by Costa et al. in 2013 [30] for high-speed rail infrastructure planning. The problem is formulated as the minimization of the infrastructure construction costs while establishing trade-offs with possible

infringement of the geometric design best practice values, overlay of sensitive land-use areas, and the value of connecting intermediate cities.

Large, restrictive values represent the design best practice for the plan view of the infrastructure's horizontal angles, termed normal horizontal angles. Smaller than normal horizontal angles are admissible when difficult circumstances are encountered, such as rough terrain or existing constructions in the deployment area. These, however, represent future limitations to the infrastructure operation phase that should be balanced with the costs of building bridges and tunnels or relocating existing infrastructure as required for implementing large horizontal angles (see Sect. 1.1.1). The model considers such trade-offs through penalties in the objective function whenever an alignment's horizontal angle is smaller than the normal one. An analogous approach is considered for the longitudinal gradient of the alignment linear sections. A penalty representing additional operation costs increases the objective function value one aims at minimizing when gradients exceed the normal gradient value. In the case of gradients, normal values are smaller than the limit ones. The negative impacts of crossing sensitive areas is also considered through a penalty in the objective function while an additional objective function term represents the overall value of connecting intermediate locations.

Constraints are defined by the regulatory safety concerns of the infrastructure layout (limit values), legislation protected land-use areas, and the obvious connection of the problem main cities. Several layers of spatial data are required that are then combined with the HSR configuration for defining the construction costs and the objective function value and verify the compliance with the problem constraints.

3.1 Implementation of the Simulated Annealing Algorithm

Implementing the simulated annealing algorithm requires that choices are made regarding the generic cooling schedule parameters but also problem-specific issues such as what a system configuration is, its objective function value, the generation of new candidate configurations, and how to derive an initial configuration [36]. These are here discussed for studies regarding a Portuguese high-speed rail network.

3.1.1 Problem-Specific Definitions

A feasible system configuration for this application is an HSR line, defined by its linear alignments, that connects the main cities Oporto, Coimbra, and Lisbon with all longitudinal gradients smaller than a regulatory limit, all horizontal angles (as a

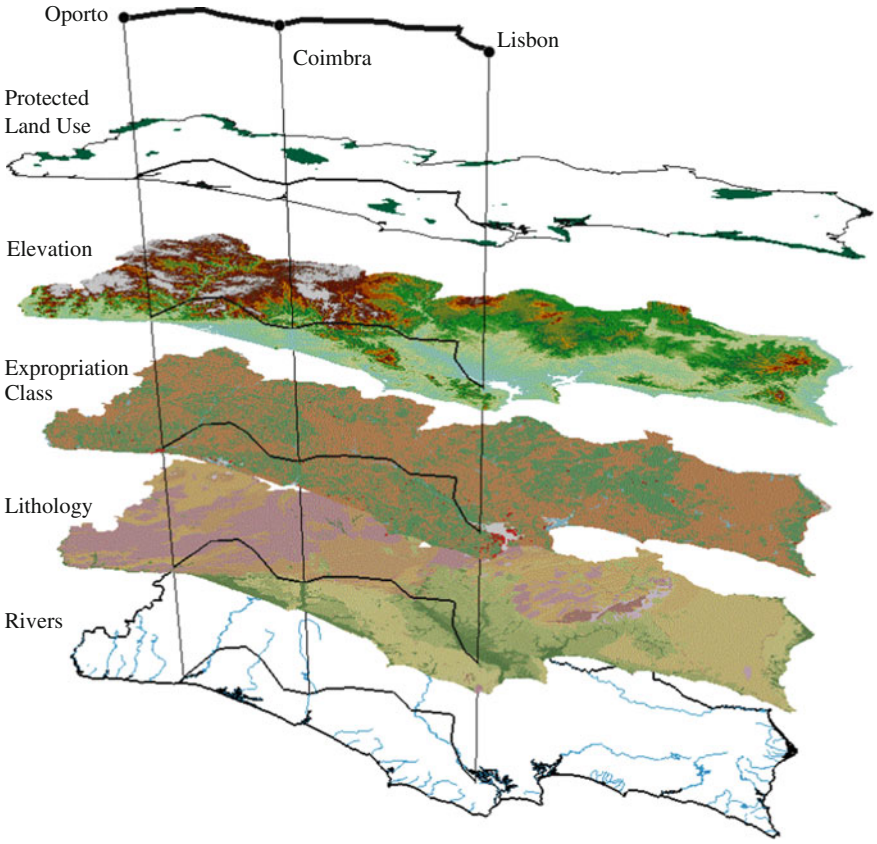

Fig. 3 The initial system configuration and the respective overlays with the problem spatial data [41]

proxy for radii) larger than a regulatory limit, and that does not overlay protected land-use areas such as natural reservations or parks. Figure 3 illustrates the data layers used [41] and the initial system configuration for implementing the algorithm.

A neighbor of any current configuration is obtained by changing, within its neighborhood structure, two randomly chosen nodes. Preliminary tests were performed considering transitions changing only one node of the HSR linear alignment. It was observed that the SAA was not able to comprehensively canvass the search space of the problem when only one node was perturbed and the final solution depended on the initial system configuration chosen.

The initial system configuration shown in Fig. 3 was arbitrarily defined. The regulatory limits for the geometry of the HSR layout are very strict due to the high running speeds of the trains. As such, randomly obtaining linear alignments connecting Oporto, Coimbra, and Lisbon that comply with layout restrictions and

additionally none of its linear sections overlays protected land-use areas adds a significant overhead to the computation time. Starting at an arbitrarily defined configuration, studies can be performed for the algorithm implementation to assess if the algorithm is able to comprehensively search the problem space, such that the final configuration is independent of the initial one.

The construction cost of any configuration is obtained by summing the costs of earthworks, expropriation, bridges, and tunnels along the infrastructure length by comparing and overlaying the HSR configuration and the local characteristics, including the influence of lithological conditions affecting the HSR cross-section [30].

3.1.2 Cooling Schedule: Equilibrium and Temperature Decrease

The studies performed show how one may define the parameters of the simulated annealing implementation governing the decrease in the temperature, the algorithm control parameter. An adaptive geometric cooling schedule is implemented based on existing research [30, 36, 42, 43].

The adaptability of the cooling schedule resides in the number of system configurations n_1 to be evaluated at each temperature step. A state of equilibrium at any given temperature step is reached if, for n_1 configurations, the optimum and the average of the objective function value do not improve. On the contrary, if either the optimum or the average does improve, other n_1 configurations are evaluated and equilibrium checked. Such implementation results in equilibrium criteria that adapt throughout the implementation stages to the search performed.

When equilibrium is reach at a given temperature, the temperature is decreased geometrically with a constant rate $0 < r < 1$. The new temperature is then given as $t_{n+1} = r \cdot t_n$.

3.1.3 Cooling Schedule: Initial and Final Temperature Criteria

The initial system temperature is defined based on the elasticity of acceptance. It represents the probability of accepting a worse new candidate configuration when at the initial system configuration c_i with objective function value E_i. The initial temperature is thus given as $t_0 = -0.1E_i/\ln(a)$. This initial system temperature allows the acceptance of a % configurations with a 10 % larger objective function value than the initial system configuration.

One final cooling schedule parameter is required, defining the termination criteria for the algorithm. The simulated annealing algorithm is terminated when reaching the termination temperature t_f that corresponds to n_2 consecutive temperature decreases without improvements observed in either the optimum or the average objective function value.

3.2 Simulated Annealing Algorithm Performance

The formulation of the cooling schedule is not specific to the problem one aims at solving and the same cooling schedule formulation may be used to solve different problems. The values of the parameters implemented for each problem, however, influence the performance of the algorithm and should be investigated. Implementations for multiple combinations of the cooling schedule parameters were studied, building on previous research [30], and the combination found to produce the best results consider the following:

- Elasticity of acceptance $a = 0.9$;
- Minimum number of iterations per temperature $n_1 = 5000$;
- Temperature decrease rate $r = 0.8$;
- Termination criterion $n_2 = 10$.

Figure 4 shows the convergence history for the solution found with the preceding cooling schedule parameters. Initially, at the early temperature steps, current system configurations with large objective function values are accepted. As the algorithm advances, the acceptance rate of worse configurations decreases and the objective function value of the current configurations converges to the optimum configuration objective function value found by the algorithm.

The maximum objective function values of current and optimum system configurations before each temperature decrease step differ and are thus plotted in

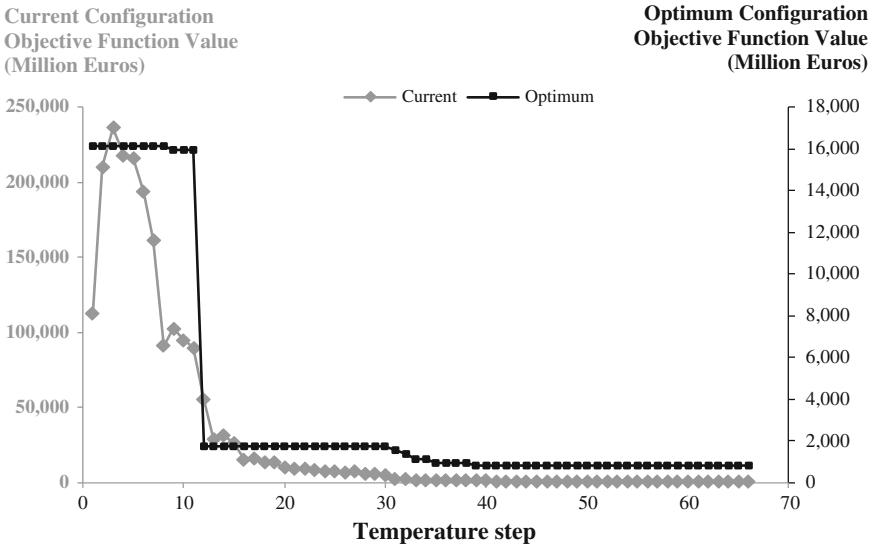

Fig. 4 Convergence history of the simulated annealing algorithm implementation: evolution of the last accepted configuration before a temperature decrease (*current*) and the best configuration found by the algorithm (*optimum*) at the time of each temperature decrease

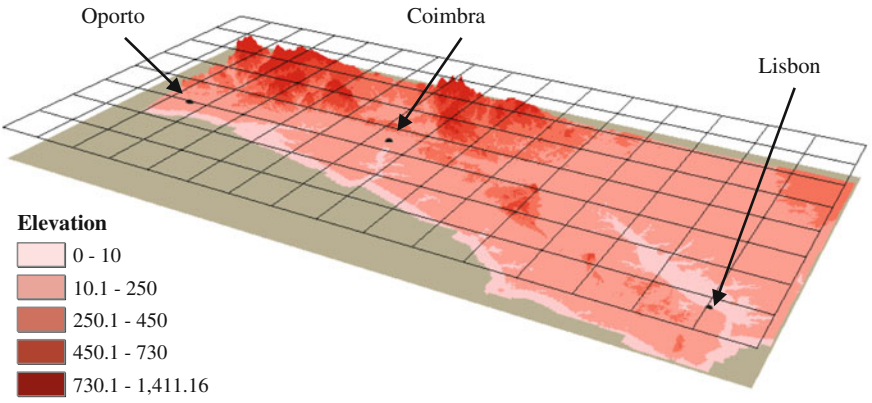

Fig. 5 Three-dimensional limits for the deployment of the high-speed rail and the ground elevation (meters)

different vertical scales. Both the current and optimum objective function values overlap at the end of the algorithm if plotted on the same vertical scale. The acceptance of configurations with large objective function values at the beginning of the algorithm is required for enabling the algorithm to globally explore the feasible search space of the problem.

One observes that the objective function value of current configurations at the early stages of the algorithm implementation is significantly larger than that of the best solution found. The relation between the high-speed rail configuration and the deployment site characteristics substantiates this observation.

The three-dimensional limits for the deployment of the infrastructure are shown in Fig. 5. A reduced rectangular area of 147.4 * 304.4 km^2 frames the plan view of the problem area and the elevation of the HSR is bounded by the depicted lower surface and the top mesh. The lower elevation limit for the configuration corresponds to 50 m below sea level and the maximum corresponds to the maximum ground elevation in the area.

It should be noted that construction costs can vary substantially within the problem space. For example, a given HSR plan view can have multiple longitudinal profiles corresponding to different extents of bridges and tunnels that, due to the large costs, can influence the overall infrastructure cost. Particularly for the cost of bridges, the cost increases significantly with the increase in the infrastructure height relatively to the ground elevation. In addition to the construction costs, one verifies if the design best practice geometry values are complied with and if the intermediate cities identified for the problem are connected. The respective values are added to the construction cost to obtain the objective function value of any given system configuration. Accordingly, high-speed rail configurations are found in the search space, which vary the objective function value in orders of magnitude.

Maier et al. in 2014 [44] discuss how large search space and complex fitness landscapes are characteristic of real-world problems. The fitness landscape depends on both the problem one aims at solving and the solving technique with the respective parameters implemented, defined by Smith et al. 2002 [45] as a multi-dimensional landscape established by the possible solutions the solving technique progresses and correspondingly mapped to the respective fitness values.

Competing objectives are also characteristic of real-world problems. For instance, in the case of high-speed rail planning, optimizing the geometry layout often leads to larger construction costs that one aims at minimizing. This fact, coupled with the computational burden arising from the complexity and size of real-world problems, makes it difficult to identify the best convergence criteria [44]. One observes in Fig. 4 that prior to the thirtieth temperature step, over ten temperature decreases occur without improvements in the best configuration found. But, as the current configurations are improving, the algorithm progresses until eventually in the thirty-first temperature step, the current optimum again improves and converges to the best configuration found. The implementation of the termination criteria should ensure that the algorithm does not terminate prematurely but also that unnecessary computations are avoided and the algorithm terminates in a finite time [44].

The construction cost of the best configuration found, which connects two intermediate stations, is in line with existing high-speed rail lines in the world [12] and shows the ability of the simulated annealing algorithm in addressing the planning of transport infrastructure. Further applications may be used to study how different preferences for conflicting objectives influence the solutions obtained and support decision making.

4 Conclusions

This chapter discusses how the simulated annealing algorithm may be used to address the planning of transportation infrastructure. The specific case of high-speed rail planning is discussed considering the location of intermediate stations, the specific requirements of the infrastructure layout, and its interrelation with the deployment site conditions. An application to a real-world case is presented that exemplifies decisions regarding the simulated annealing implementation. These decisions are discussed in light of existing research.

Ultimately, solving such real-world problems aims at supporting decision making. This may be influenced by political decisions and based on the different perspectives of multiple stakeholders. Providing only a single best solution for a particular set of conditions is not adequate for decision making. The application of the optimization techniques for real-world problems should thus support the investigation of multiple opportunities and preferences for the problem at hand.

Acknowledgments The authors would like to acknowledge the financial support of *Fundação para a Ciência e Tecnologia* (FCT) through doctoral grant (SFRH/BD/43012/2008) and the access to preliminary studies provided by former *Rede Ferróviaria de Alta Velocidade, S.A.* (RAVE).

References

1. Givoni, M., Rietveld, P.: Do cities deserve more railway stations? The choice of a departure railway station in a multiple-station region. J. Transp. Geogr. **36**, 89–97 (2014). doi:10.1016/j.jtrangeo.2014.03.004
2. Brons, M., Givoni, M., Rietveld, P.: Access to railway stations and its potential in increasing rail use. Transp. Res. Part A Policy Pract. **43**, 136–149 (2009). doi:10.1016/j.tra.2008.08.002
3. Givoni, M., Rietveld, P.: The access journey to the railway station and its role in passengers' satisfaction with rail travel. Transp. Policy **14**, 357–365 (2007). doi:10.1016/j.tranpol.2007.04.004
4. Repolho, H.M., Antunes, A.P., Church, R.L.: Optimal location of railway stations: the Lisbon-Porto high-speed rail line. Transp. Sci. **47**, 330–343 (2013). doi:10.1287/trsc.1120.0425
5. Vickerman, R.: High-speed rail in Europe: experience and issues for future development. Ann. Reg. Sci. **31**, 21–38 (1997). doi:10.1007/s001680050037
6. Levinson, D.M.: Accessibility impacts of high-speed rail. J. Transp. Geogr. **22**, 288–291 (2012). doi:10.1016/j.jtrangeo.2012.01.029
7. EC: Commission Decision of 20 December 2007 concerning a technical specification for interoperability relating to the infrastructure sub-system of the trans-European high-speed rail system. Off. J. Eur. Union (2008)
8. CEN: Railway application—track alignment design parameters—track gauges 1435 mm and wider—Part 1: plain line. ENV 13803-1, CEN—European Committee for Standardization (2002)
9. RTRI: Design Standards for Railway Structures—Displacement Limits (2007)
10. Levinson, D., Gillen, D., Kanafani, A., Mathieu, J.: The Full Cost of Intercity Transportation—A Comparison of High Speed Rail, Air And Highway Transportation In California (1996)
11. Levinson, D., Mathieu, J.M., Gillen, D., Kanafani, A.: The full cost of high-speed rail: an engineering approach. Ann. Reg. Sci. **31**, 189–215 (1997). doi:10.1007/s001680050045
12. Campos, J., de Rus, G.: Some stylized facts about high-speed rail: a review of HSR experiences around the world. Transp. Policy **16**, 19–28 (2009). doi:10.1016/j.tranpol.2009.02.008
13. Jong, J.C.: Optimizing highway alignments with genetic algorithms. Ph.D. Dissertation. Department of Civil and Environmental Engineering (1998)
14. Jong, J.C., Jha, M.K., Schonfeld, P.: Preliminary highway design with genetic algorithms and geographic information systems. Comput. Civ. Infrastruct. Eng. **15**, 261–271 (2000)
15. Cheng, J.F., Lee, Y.S.: Model for three-dimensional highway alignment. J. Transp. Eng. **132**, 913–920 (2006). doi:10.1061/(asce)0733-947x(2006)132:12(913)
16. Lee, Y., Tsou, Y.R., Liu, H.L.: Optimization method forhighway horizontal alignment design. J. Transp. Eng. **135**, 217–224 (2009). doi:10.1061/(ASCE)0733-947X(2009)135:4(217)
17. Jha, M.K.: Criteria-based decision support system for selecting highway alignments. J. Transp. Eng. **129**, 33–41 (2003). doi:10.1061/(asce)0733-947x(2003)129:1(33)
18. Jha, M.K., Schonfeld, P.: A highway alignment optimization model using geographic information systems. Transp. Res. Part A Policy Pract. **38**, 455–481 (2004). doi:10.1016/j.tra.2004.04.001
19. Kim, E., Jha, M.K., Schonfeld, P., Kim, H.S.: Highway alignment optimization incorporating bridges and tunnels. J. Transp. Eng. **133**, 71–81 (2007). doi:10.1061/(asce)0733-947x(2007)133:2(71)

20. Fwa, T.F., Chan, W.T., Sim, Y.P.: Optimal vertical alignment analysis for highway design. J. Transp. Eng. **128**, 395–402 (2002). doi:10.1061/(asce)0733-947x(2002)128:5(395)
21. Kim, E., Jha, M.K., Schonfeld, P.: Intersection construction cost functions for alignment optimization. J. Transp. Eng. **130**, 194–203 (2004). doi:10.1061/(asce)0733-947x(2004)130:2(194)
22. Gipps, P.G., Gu, K.Q., Held, A., Barnett, G.: New technologies for transport route selection. Transp. Res. Part C Emerg. Technol. **9**, 135–154 (2001). doi:10.1016/S0968-090X(00)00040-1
23. Jong, J.C., Schonfeld, P.: Cost functions for optimizing highway alignments. Transp. Res. Rec. J. Transp. Res. Board **1959**, 58–67 (1999). doi:10.3141/1659-08
24. Parker, N.A.: Rural highway route corridor selection. Transp. Plan Technol. **3**, 247–256 (1977). doi:10.1080/03081067708717111
25. Trietsch, D.: A family of methods for preliminary highway alignment. Transp. Sci. **21**, 17–25 (1987)
26. Chew, E.P., Goh, C.J., Fwa, T.F.: Simultaneous optimization of horizontal and vertical alignments for highways. Transp. Res. Part B Methodol. **23**, 315–329 (1989). doi:10.1016/0191-2615(89)90008-8
27. Kang, M.W., Jha, M.K., Schonfeld, P.: Applicability of highway alignment optimization models. Transp. Res. Part C Emerg. Technol. **21**, 257–286 (2012). doi:10.1016/j.trc.2011.09.006
28. Shafahi, Y., Bagherian, M.: A customized particle swarm method to solve highway alignment optimization problem. Comput. Civ. Infrastruct. Eng. **28**, 52–67 (2013). doi:10.1111/j.1467-8667.2012.00769.x
29. Angulo, E., Castillo, E., Garcia-Rodenas, R., Sanchez-Vizcaino, J.: Determining highway corridors. J. Transp. Eng. **138**, 557–570 (2012). doi:10.1061/(ASCE)TE.1943-5436.0000361
30. Costa, A.L., Cunha, C., Coelho, P.A.L.F., Einstein, H.H.: Solving high-speed rail planning with the simulated annealing algorithm. J. Transp. Eng. **139**, 635–642 (2013). doi:10.1061/(ASCE)TE.1943-5436.0000542
31. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., et al.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087–1092 (1953)
32. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
33. Cerny, V.: Thermodynamical approach to the traveling salesman problem—an efficient simulation algorithm. J. Optim. Theory Appl. **45**, 41–51 (1985)
34. Aarts, E., Korst, J., Van Laarhoven, P.J.M.: Simulated annealing. In: Aarts, E., Lenstra, J.K. (eds.) Local Search in Combinatorial Optimization, 1st edn., pp 91–120. Wiley, New York (1997)
35. Dekkers, A., Aarts, E.: Global optimization and simulated annealing. Math. Program. **50**, 367–393 (1991)
36. Johnson, D.S., Aragon, C.R., MCGeoch, L.A., Schevon C.: Optimization by simulated annealing—an experimental evaluation. Part 1. Graph Partit. Oper. Res. **37**, 865–892 (1989) doi:10.1287/opre.37.6.865
37. Osman, I.H., Laporte, G.: Metaheuristics: a bibliography. Ann. Oper. Res. **63**, 513–623 (1996)
38. Bertsimas, D., Nohadani, O.: Robust optimization with simulated annealing. J. Glob. Optim. **48**, 323–334 (2010). doi:10.1007/s10898-009-9496-x
39. Jilla, C.D., Miller, D.W.: Assessing the performance of a heuristic simulated annealing algorithm for the design of distributed satellite systems. Acta Astronaut. **48**, 529–543 (2001)
40. Kirkpatrick, S.: Optimization by simulated annealing: quantitative studies. J. Stat. Phys. **34**, 975–986 (1984). doi:10.1007/BF01009452
41. APA: Environmental Atlas. In: Port. Environ. Agency (2012). http://sniamb.apambiente.pt/webatlas/
42. Cunha, M.C.: On solving aquifer management problems with simulated annealing algorithms. Water Resour. Manag. **13**, 153–169 (1999)

43. Cunha, M.C., Sousa, J.J.O.: Hydraulic infrastructures design using simulated annealing. J. Infrastruct. Syst. **7**, 32–39 (2001)
44. Maier, H.R., Kapelan, Z., Kasprzyk, J., et al.: Evolutionary algorithms and other metaheuristics in water resources: current status, research challenges and future directions. Environ. Model Softw. **62**, 271–299 (2014). doi:10.1016/j.envsoft.2014.09.013
45. Smith, T., Husbands, P., O'Shea, M.: Fitness landscapes and evolvability. Evol. Comput. **10**, 1–34 (2002). doi:10.1162/106365602317301754

A Hybrid Bat Algorithm with Path Relinking for the Capacitated Vehicle Routing Problem

Yongquan Zhou, Qifang Luo, Jian Xie and Hongqing Zheng

Abstract The capacitated vehicle routing problem (CVRP) is an NP-hard problem with both engineering and theoretical interests. In this paper, a hybrid bat algorithm with path relinking (HBA-PR) is proposed to solve CVRP. The HBA-PR is constructed based on the framework of the continuous bat algorithm, the greedy randomized adaptive search procedure (GRASP) and path relinking are effectively integrated into the bat algorithm. Moreover, in order to further improve the performance, the random subsequences and single-point local search are operated with certain loudness (a probability). In order to verify the effectiveness of our approach and its efficiency and compare with other existing methodologies, several classical CVRP instances from three classes of CVRP benchmarks are selected to test. Experimental results and comparisons show the HBA-PR is effective for solving CVRPs.

Keywords Bat algorithm · Capacitated vehicle routing problem · Path relinking · GRASP · Metaheuristic algorithm

1 Introduction

The vehicle routing problem (VRP) is a classical combinatorial optimization problem that was proposed in the late 1950s and it is still a research hotspot in the field of Operations Research. The capacitated vehicle routing problem (CVRP) was introduced by Dantzig and Ramser in 1959 [1], which concerns that a set of customer demands have to be served with a fleet of vehicles from a depot or central node, each

Y. Zhou (✉) · Q. Luo · J. Xie · H. Zheng
College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China
e-mail: yongquanzhou@126.com

Y. Zhou
Guangxi High School Key Laboratory of Complex System and Computational Intelligence, Nanning 530006, China

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*, Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_12

vehicle has the uniform capacity and each customer has a certain demand that must be satisfied at the minimal cost. These costs usually represent the distances, traveling times, number of vehicles employed or a combination of these factors.

It is well-known that the CVRP is an NP-hard problem [2]. Various approaches have been presented to solve the CVRP during the last decades, such as linear programming [3], several metaheuristics [4–6], and many hybrid heuristics with variable neighborhood search or constructive heuristic methods [7–10]. The overview of methods presented in [6] shows that at least 29 different methods for solving CVRP exist, and all achieved more or less comparable performance. Although several methods can produce good solutions, the computational time is long when the scale of instances is large. Meanwhile, an abundance of methods for the CVRP are population-based algorithms and the parameter setting of an algorithm is important, however, the parameter settings of many metaheuristics have not been considered in the literature.

The bat algorithm (BA) is a fairly new metaheuristic proposed by Yang [11, 25] in 2010, which inspired by the intelligent echolocation behavior of micro-bats when they forage. As we know, many new metaheuristics have been widely used and successfully applied to solve the CVRP, though BA has not yet been applied to solve the CVRP. However, BA has been applied to solve other problems with great success. For example, Gandomi et al. focus on solving constrained optimization tasks [12]. Yang and Gandomi apply bat algorithm to solve many global engineering optimizations [13]. Mishra et al. proposed a model for classification using the bat algorithm to update the weights of a functional link artificial neural network (FLANN) classifier [14]. Meanwhile, some researchers have improved the bat algorithm and applied it to various optimization problems. Xie et al. proposed a DLBA bat algorithm based on differential operator and Lévy flights trajectory to solve function optimization and nonlinear equations [15]. Wang et al. proposed a new bat algorithm with mutation (BAM) to solve the uninhabited combat air vehicle (UCAV) path planning problem [16]. In this paper, we propose a hybrid bat algorithm (HBA-PR) to solve the capacitated vehicle routing problem.

The rest of this paper is organized as follows. The problem of CVRP and the original bat algorithm are described in Sect. 2. The hybrid bat algorithm (HBA-PR) for CVRP is described in Sect. 3. The experimental results of the HBA-PR and comparisons with other previous algorithms are shown in Sect. 4. In the last section, we conclude this paper and point out some future work in Sect. 5.

2 Problem Descriptions and Bat Algorithm

2.1 Capacitated Vehicle Routing Problem

The CVRP is considered to be the classical version of the VRP which designs a set of customer demands that have to be served with a fleet of vehicles from a depot or central node, each vehicle has a uniform capacity, and each customer has a certain

demand; the objective is to make the expended cost as low as possible. Let $G = (V, E)$ be a complete graph with a set of vertices $V = \{0, 1, \ldots, k\}$, where the vertex $\{0\}$ represents the depot and the remaining ones are the customers. Each edge $e_{ij} = \{i, j\} \in E$ has a non-negative cost c_{ij} and each customer $i \in V' = V \backslash \{0\}$ has a demand d_i. Let $S = (1, 2, \ldots, m)$ be the set of homogeneous vehicles with capacity Q. The CVRP consists of constructing a set of up to k routes in such a way that: (1) every route starts and ends at the depot; (2) all demands are accomplished; (3) the vehicle's capacity is not exceeded; (4) a customer is visited by only a single vehicle; and (5) the sum of costs is minimized. The mathematical formulas are defined as follows [10].

$$\min Z = \sum_{i=0}^{k} \sum_{j=0}^{k} \sum_{s=0}^{m} c_{ij} e_{ijs} \tag{1}$$

$$\begin{aligned} \text{s.t.} \sum_{i=0}^{k} d_i y_{is} \leq Q, \quad & \text{s} = 1,2,\ldots m; \\ \sum_{s=1}^{m} y_{is} = 1, \quad & \text{i} = 1,2,\ldots, k; \\ \sum_{i=0}^{k} e_{ijs} = y_{is}, \quad & j = 1, 2, \ldots, k; \quad s = 1, 2, \ldots, m; \\ \sum_{j=0}^{k} e_{ijs} = y_{is}, \quad & i = 1, 2, \ldots, k; \quad s = 1, 2, \ldots, m \end{aligned} \tag{2}$$

where s denotes the number of vehicle, $e_{ijs} = 1$ if vehicle s from i to j, otherwise $e_{ijs} = 0$; In addition, $y_{is} = 1$ if vehicle s is loading (active), otherwise $y_{is} = 0$.

2.2 Basic Bat Algorithm

The basic bat algorithm (BA) is a metaheuristic, first introduced by Yang in 2010. In simulations of BA, under several idealized rules, the updated rules of bats' positions x_i and velocities v_i in a D-dimensional search space are defined. The new solutions x_i^t and velocities v_i^t at generation t are given by

$$\begin{aligned} f_{r_i} &= f_{r_{\min}} + (f_{r_{\max}} - f_{r_{\min}})\beta \\ v_i^t &= v_i^{t-1} + (x_i^{t-1} - x_*) f_{r_i} \\ x_i^t &= x_i^{t-1} + v_i^t \end{aligned} \tag{3}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution, fr_i denotes frequency of each bat. Generally, the frequency is within a range, that is

$fr_i \in [fr_{\min}, fr_{\max}]$. Here x_* is the current global best location (solution) which is located after comparing all the solutions among all the n bats at each iteration.

After the position updating of bats, a random number is generated, if the random number is greater than the pulse emission rate r_i, a new position will be generated around the current best solutions, and it can be represented by (4)

$$x = x_* + \varepsilon \times Ld_t, \tag{4}$$

where $\varepsilon \in [-1, 1]$ is a random number, while $Ld_t = \langle Ld_i^t \rangle$ is the average loudness of all the bats at current generation t.

Furthermore, the loudness Ld_i and the pulse emission rate r_i will be updated and a solution will be accepted if a random number is less than loudness Ld_i and $f(x_i) < f(x_*)$. Ld_i and r_i are updated by (5).

$$Ld_i^{t+1} = \alpha \times Ld_i^t,\ r_i^{t+1} = r_i^0 \times [1 - \exp(-\gamma \times t)], \tag{5}$$

where α, γ are constants, and $f(\bullet)$ is the fitness function. The algorithm repeats until the termination criterion is met.

3 Hybrid Bat Algorithm with Path Relinking for CVRP

3.1 Solution Representation in HBA-PR

Since the standard BA is a continuous optimization algorithm, the standard continuous encoding scheme of BA cannot be used to solve CVRP directly. In order to apply BA to solve CVRP, the first step is to devise a suitable representation for the candidate solutions in designing a hybrid bat algorithm for a particular problem. Each individual is a sequence with integer numbers which are the order of visiting these customers where the number 0 represents the depot. For example, if we have the following three routes (Fig. 1):

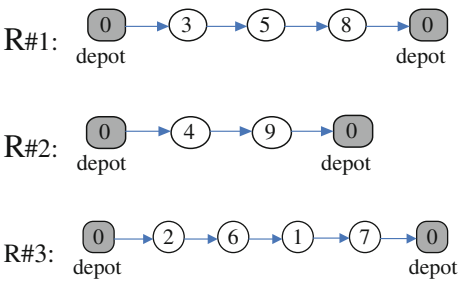

Fig. 1 Individuals are a sequence with integer numbers

Where the coded individual with integers is 0 → 3 → 5 → 8 → 4 → 9 → 2 → 6 → 1 → 7 → 0, and the bat individual is represented as 3 → 5 → 8 → 4 → 9 → 2 → 6 → 1 → 7.

3.2 Hybrid Bat Algorithm

Aimed at the capacitated vehicle routing problem, based on the idea of the bat algorithm, a hybrid bat algorithm is proposed here, which integrates greedy randomized adaptive search procedure (GRASP) heuristic and the bat algorithm, and path relinking as an intensification strategy to explore local trajectories connecting elite solutions obtained by the proposed algorithm. The hybrid bat algorithm with path relinking is named as HBA-PR.

GRASP [17, 18] is a heuristic that has already been applied to many optimization problems successfully [19–21]. GRASP consists of a two phase iterative process: a construction phase and a local search phase. In the first phase, a greedy randomized solution is built. Since this solution is not guaranteed to be locally optimal, a local search is performed in the second phase. The final result is simply the best solution found over all iterations.

The construction phase can be described as a process which stepwisely adds one element at a time to a partial (incomplete) solution. According to a greedy function, all elements are sorted, and the Restricted Candidate List (RCL) is constructed based on the order; and then selects element randomly from this list. In the second phase, a local search is initialized from these points, this iterative process is repeated until a termination criterion is met and the best solution found over all iterations is taken as the result.

RCL is created using a parameter α to restrict the size of this list. Candidate $e \in C$ is sorted according to their greedy function value $f(e)$. In a cardinality-based RCL, the latter is made up by the k top-ranked elements. In a value-based construction, the RCL consists of the elements in the set

$$\{e \in C : f_* \leq f(e) \leq f_* + \alpha \times (f^* - f_*)\} \tag{6}$$

where $f_* = \min(f(e) : e \in C), f^* = \max\{f(e) : e \in C\}$, and $\alpha \in [0, 1]$. Since the best value for α is often difficult to determine, a random value is often assigned. The values for α adopted in the constructive heuristics are set using reactive strategies, which usually leads to better performance than using fixed values [22].

In the original bat algorithm, the bat individual randomly selects a certain range of frequency, and its velocity is updated according to their selected frequency, at last, a new position is generated using its velocity and its own position. The idea is that the position of bat individual is updated by adjusting its frequency of sonic pulses. In this paper, the position updating of bat adopted GRASP to generate a new

position, the frequency is used for restricting the size of CRL, frequency equivalent to parameter α in GRASP, and the frequency is a variable value. In the basic bat algorithm, the loudness *Ld* and the pulse emission rate *r* are updated accordingly as the iterations proceed. As the loudness usually decreases, while the rate of pulse emission increases, it indicates the bats are approaching their prey (optimum solution). The pulse emission rate *r* is updated by (7)

$$r(t) = (1 + \exp(5/t_{\max} - \times(t - t_{\max}/2))^{-1}, \tag{7}$$

where *t* denotes the *t*th generation, $t_{\max}$ is the maximal generation. The rate *r* is similar to Sigmoid function, the frequency *fr* is determined according to the pulse emission rate *r*, which is represented by (8)

$$fr = \begin{cases} 1 - \max(0.2, \min(0.8, r)), & \text{rand} > r, \\ \max(0.2, \min(0.8, r)), & \text{rand} \le r, \end{cases} \tag{8}$$

where *rand* is a random number, and 0.2 and 0.8 are empirical parameters reference values [18]. The frequency *fr* decreases gradually at first, and then increases gradually while the generation *t* increases. Figure 2 is the changing curve of rate *r*, and Fig. 3 is an example of frequency *fr*, and Algorithm 1 shows the pseudo-code of greedy randomized construction with frequency *fr*.

Algorithm1. ***Greedy_Randomized_Construction*** *(fr)*

$S \leftarrow$ *Rrand_Chosen_ Vertex(v); // Rrandomly chosen a vertex as initial solution*
  $C \leftarrow V \setminus v$; *// The candidate set C is initialized*
  While $C \neq \varnothing$ ***do***
    $IC \leftarrow$ *Evaluate_Incremental_Costs(S); // Incremental costs are evaluated*
    $ic_{\min} \leftarrow \min(\{ic \in IC\})$;
    $ic_{\max} \leftarrow \max(\{ic \in IC\})$;
    $RCL \leftarrow \{e \in C \mid ic(e) \le ic_{\min} + fr \times (ic_{\max} - ic_{\min})\}$; *// RCL is created*
    $s \leftarrow$ *Select_Element (RCL); // a vertex s is randomly selected from RCL*
$S \leftarrow$ *Obtain_Min_Solution(S,s); // partial tour is updated by inserting vertex s*
    $C \leftarrow C \setminus \{s\}$; *// the candidate set C is updated*
  end
  return *S*

The local search phase uses the 2-Opt heuristic for exchanging. Algorithm 2 shows the local search procedure. The input parameter is an initial solution *S* obtained by the Greedy_Randomized_Construction procedure. The current route needs to divide into *m* sub-routes according to the load, the start, and the end of sub-route is represented by 0, for example, $S = \{1, 2, 3, 4, 5, 6, 7\}$, sub-route is $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0$, and $0 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 0$.

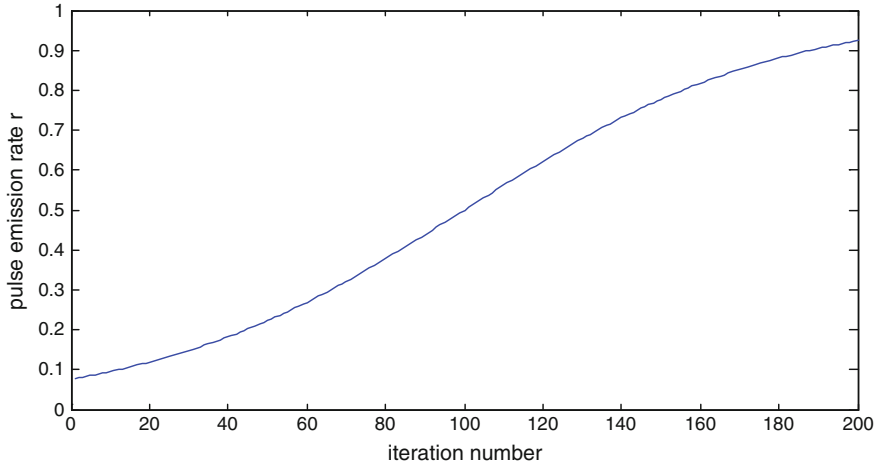

Fig. 2 Changing curve of rate *r*

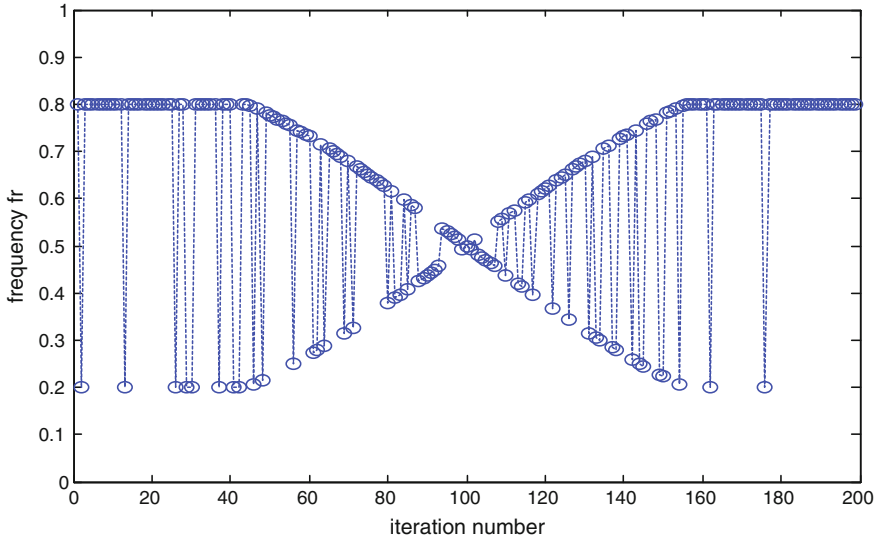

Fig. 3 Changing curve of frequency *fr*

Algorithm2. ***Local_Search_Phase*** *(S)*

```
S′ ← S ;
{sr} ← Construct_Sub-route( S′ );
for each sub-route sr do
  sr′ ← 2-opt ( sr );  // carry out the 2-opt operation
end
S′ ← Construct_Indiviaual( {sr′} );
if  f (S′) < f (S) then
  S ← S′
end
return  S
```

3.3 Hybrid Bat Algorithm with Path Relinking

Path relinking was originally proposed by Glover [23], Laguna and Martí [24] were the first to use path relinking within a GRASP strategy. Path relinking generates new solutions by exploring trajectories connecting an initial solution x_s to an elite guiding solution x_t. The path relinking procedure consists of selecting moves that introduce attributes contained in the guiding solution x_t to the initial solution x_s until the initial solution is completely transformed in the guiding solution x_t. Path relinking may also be viewed as a constrained local search strategy applied to the initial solution x_s. Furthermore, there are several alternatives that have been considered, which involve some trade-offs between the computation time and solution quality. These alternatives include periodical relinking, forward relinking, backward relinking, back and forward relinking, mixed relinking, randomized relinking, and truncated relinking [18].

One important issue in implementing a path relinking technique is the strategy to construct the elite set (ES). We adopted a fixed-size elite set and a solution x is inserted into the ES as follows.

A solution x is always inserted into ES if it is not full. Otherwise, the generated solution x is inserted in ES only if its cost is better than the worst cost solution found in ES, and the worst cost solution is replaced by the solution x. Algorithm 3 shows the pseudo-code for the algorithm to construct and maintain the elite set ES.

Algorithm 3. ***Elite _Set(x)***

if $cardinality(ES) \neq ps$ ***then*** *// ps is the population size*

$ES \leftarrow ES \bigcup \{x\}$

else

$ES_w \leftarrow$ *the worst solution in ES*

if $f(x) < f(ES_w)$ ***then***

$ES \leftarrow ES \setminus \{ES_w\}$

$ES \leftarrow ES \bigcup \{x\}$

end

end

return *ES*

Algorithm 4 shows the pseudo-code of path relinking procedure, an instance is $x_s = \{1,2,3,4,5,6,7\}$, $x_t = \{1,3,4,2,5,6,7\}$. According to the main process of Algorithm 4, the first different element is position 2 in x_s, after executing a replacement operation, the new solution becomes $x_1 = \{1,3,4,2,5,6,7\}$; the second different element is position 3 in x_s, after executing the replacement operation, the new solution is $x_2 = \{1,2,4,3,5,6,7\}$. The third different element is position 4 in x_s, after executing the replacement operation, then x_3. If x_2 is only one better than x_t, then x_2 will be returned.

Algorithm 4. ***Path_Relinking(ES_* , x);***

$x_s \leftarrow x$; *// initial solution*

$x_t \leftarrow ES_*$; *// guiding solution*

$f(x_t) \leftarrow f(ES_*)$;

$\Delta \leftarrow$ *Difference(* x_s, x_t *); // find out the difference position between* x_s x_t

for *i=1: cardinality*(Δ) ***do***

$j \leftarrow$ *Find_Position(* x_s, x_t, Δ_i *); // find the position of the element of* x_t *in* Δ_i *which in* x_s

$x_i \leftarrow$ *Replace(* x_s, Δ_i, j *); // replace the j position of* x_s *with the element of* x_s *in* Δ_i

$x_i \leftarrow$ *Replace(* x_s, x_t, Δ_i *); //replace the* Δ_i *position of* x_s *with the corresponding element of* x_t

if $f(x_i) < f(x_t)$ ***then***

$x_* \leftarrow x_i$;

$f(x_t) \leftarrow f(x_i)$;

end

end

return x_*

3.4 Subsequence and Single-Point Local Search

In this paper, the loudness Ld_i of bat individual i is related to its own fitness fit_i, the better the fitness, and the lower the loudness. The loudness can be described by (9)

$$Ld_i = (\text{fit}_i - \text{fit}_* + 0.1)/(\text{fit}^* - \text{fit}_* + 0.1), \tag{9}$$

where constant 0.1 is used for avoiding the denominator being zero. Here, fit_i is the fitness of individual i, fit_* and fit^* are the minimum and maximum fitnesses in current population, respectively. In HBA-PR, the loudness reflects the quality of an individual. In this algorithm, there are two kinds of local searches embedded into HBA-PR to further improve the performance, and they are the random subsequence local search and the single-point local search. The random subsequence local search includes random subsequence inverse and random subsequence insertion, and the single-point local search includes single-point insert and single-point swap.

For a random subsequence insert, an origin of subsequence is randomly selected at first, and then a length of subsequence is randomly selected, which is less than the length of individual S. Second, after determining the subsequence $S1$, a random insertion point is selelcted in the remainder of subsequence $S2$, $S = S1 \cup S2$, and then $S1$ is inserted into $S2$ location in the insert point. An example is shown in Fig. 3. For the random subsequence inverse, a subsequence is randomly chosen with a random length, and then the inverse operation is performed. An example is shown in Fig. 4.

For a single-point swap, two different positions are chosen from a permutation randomly and are then swapped (Fig. 5). For a single-point insertion, two different positions are chosen from a permutation randomly and the element in the first position is inserted into the back of second element. Similarly, two instances are shown in Figs. 6 and 7.

In the local search part, the random subsequence local search is performed before random single-point local search. The random subsequence insert and the random subsequence inverse operations are preformed according to loudness Ld. In other words, if a random number is greater than the loudness Ld_i, the random subsequence insert is performed; otherwise, the random subsequence inverse is performed. Similarly, the random single-point insert and the random single-point swap operations are preformed with the loudness Ld. If a random number is greater than the loudness Ld_i, the insert operation is performed; otherwise, the swap operation is performed. Note that, where a local search is operated on the current optimal individual ES_* in elite set ES, and a local search is performed for each individual. Algorithms 5 and 6 show the pseudocodes of the subsequence and single-point local search.

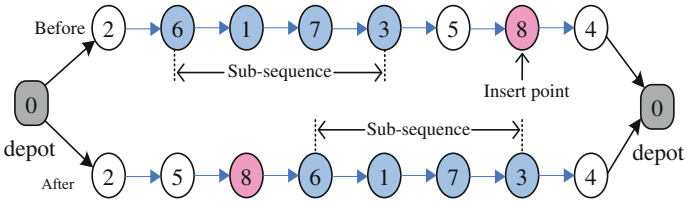

Fig. 4 Random subsequence insert operation

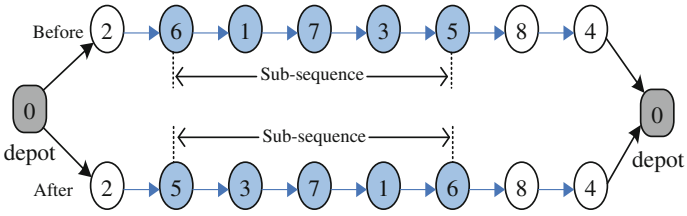

Fig. 5 Random subsequence inverse operation

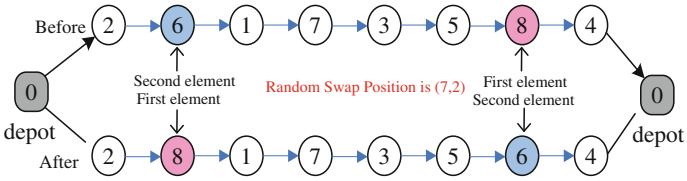

Fig. 6 Single-point swap operation

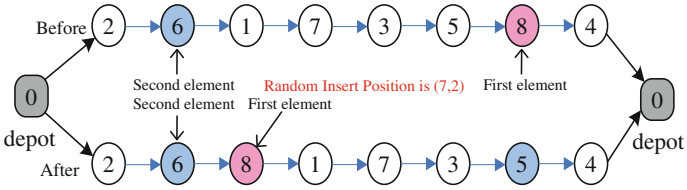

Fig. 7 Single-point insert operation

Algorithm 5. ***Sub-sequence_ Local_Search*** (ES_*)

```
for i=1:ps do
  if rand> Ld_i then
      x ← Sub-sequence_Insert( ES_* );    // perform random sub-sequence insert operation
  else
      x ← Sub-sequence_ Inverse( ES_* );  // perform random sub-sequence inverse operation
  end
end
return x
```

Algorithm 6. ***Single-point_Local_Search*** (ES_*)

```
for i=1:ps do
  if rand> Ld_i then
x ← Single-point_Insert( ES_* );  // perform random single-point insert operation
  else
x ← Single-point_Swap( ES_* );  // perform random single point swap operation
  end
end
return x
```

3.5 HBA-PR Framework for CVRP

We propose in this work to incorporate the greedy randomized adaptive search procedure, path relinking strategies, subsequence, and single-point local search to the bat algorithm by defining distinct ways to solve the capacitated vehicle routing problem. This iterative process is repeated until the termination criterion is met, Algorithm 7 shows the pseudocode of HBA-PR for CVRP.

Algorithm 7. ***HBA-PR***

```
Initialize the ps ,bat population and other parameters ;
Evaluate fitness for each individual;
ES ← Elite _Set( x );
while t<tmax do
  Compute pulse emission rate by (6);
 Determine frequency fr by (7);
 for i=1:ps do
      x ← Greedy_Randomized_Construction( fr );
      x ← Local_Search_Phase ( x );
 end
 Evaluate fitness for each individual x ;
 ES ← Elite _Set( x );
 ES* ← Select_Best_Elite( ES );  // select the best individual in elite set ES
 for i=1:ps do
      x ← Path_Relinking( ES* , x);
 end
 Evaluate fitness for each individual x ;
 ES ← Elite _Set( x );
 Ld_i ← Compute_Loudness( f(x) ) ;  //Compute loudness of each individual by (8);
 ES* ← Select_Best_Elite( ES );
 x ← Sub-sequence_ Local_Search ( ES* ) //carry out random sub-sequence local search
 Evaluate fitness for each individual x ;
 ES ← Elite _Set( x );
  ES* ← Select_Best_Elite( ES );
 x ← Single-point_Local_Search ( ES* ) //carry out random single-point local search
 Evaluate fitness for each individual x ;
 ES ← Elite _Set( x );
 t=t+1
end
Output result and plot
```

4 Numerical Simulation Results and Comparison

The performance of the proposed HBA-PR is extensively tested by a large number of experimental studies; computational simulations are carried out with some well-studied problems taken from the web http://www.branchandcut.org/, a reference site which contains some detailed information regarding a large number of benchmark instances. In this paper, 12 instances from three classes of benchmarks are selected. The first class is Augerat et al. (Set A) instances. The second class is Augerat et al. (Set P) instances, and the third class is Augerat et al. (Set E) instances. So far, these problems have been widely used as the benchmarks to validate the performance of algorithms by many researchers.

All computational experiments are conducted with MATLAB 2012a, and in our simulations, numerical experiments are run on a PC with AMD Athlon(tm) II X4 640 Processor 3.0 GHz and 2.0 GB memory. In the experiment, the termination criterion is set as the maximum generation of $t_{\max} = 200$. Each instance is independently run 15 times for comparison.

4.1 Parameter Analysis

In the subsection, the parameters of HBA-PR are determined by experiments, and the impact of each parameter is analyzed. In particular, the HBA-PR has few parameters; we only need to test population size *ps* in HBA-PR. A small *ps* may lead to insufficient population information, and the diversity cannot be guaranteed. On the other side, a large one indicates that diversity is sufficient, but the computing time will increase and the precision of the optimal solution may have lesser improvement. In order to evaluate the sensitivity of parameters *ps*, three benchmarks selected from different benchmark set are chosen to run 10 times. These benchmarks are *A_n33_k5, E_n23_k3, and P_n19_k2*, and the statistical result and convergence curves are shown in Figs. 8, 9, 10 and 11.

The ordinate normalized fitness (log) in Figs. 8, 9 and 10 is the logarithm of normalized fitness; the aim is to show the convergence curves clearly. The normalization formula of fitness is $(\text{fit} - \text{fit}_*)/(\text{fit}^* - \text{fit}_*)$, where fit_* is the best-known solution and fit^* is the initial fitness.

Figure 8 represents the relative error of test case *A_n33_k5, E_n23_k3 and P_n19_k2* after 10 times independent runs, which shows that the sensitivity of parameter *ps*. From the three test cases, the parameter *ps* can be determined. For *A_n33_k5* and *P_n19_k2*, the performance of HBA-PR is better when *ps* = 30. For *E_n23_k3*, *ps* = 50 is best, but the performance is good as well while *ps* = 30. From Figs. 8, 9 and 10, the information provided by population is sufficient when *ps* is greater than 20; however, the convergence rate is better when *ps* = 30 among the three instances. Considering the tradeoff between the stability of the algorithm and the rate of convergence, the parameter *ps* took a compromising value, *ps* = 30.

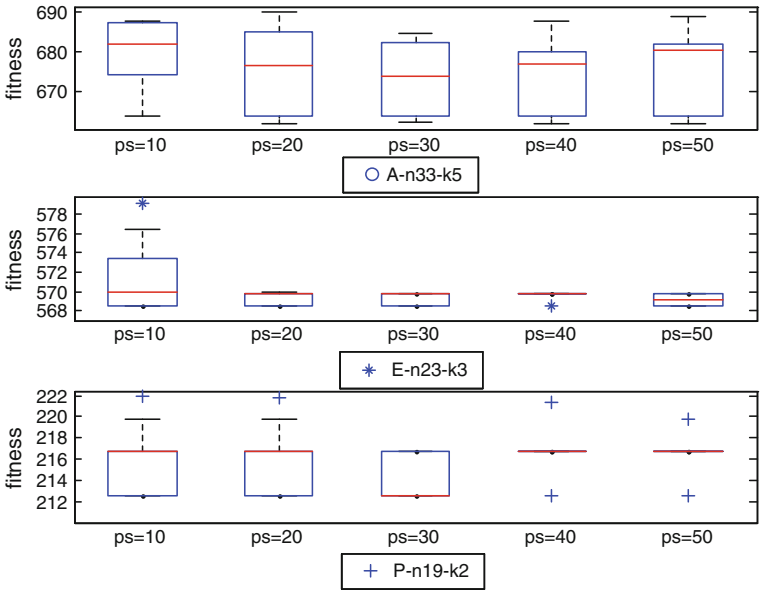

Fig. 8 Box-and-whisker diagram of selected three instances

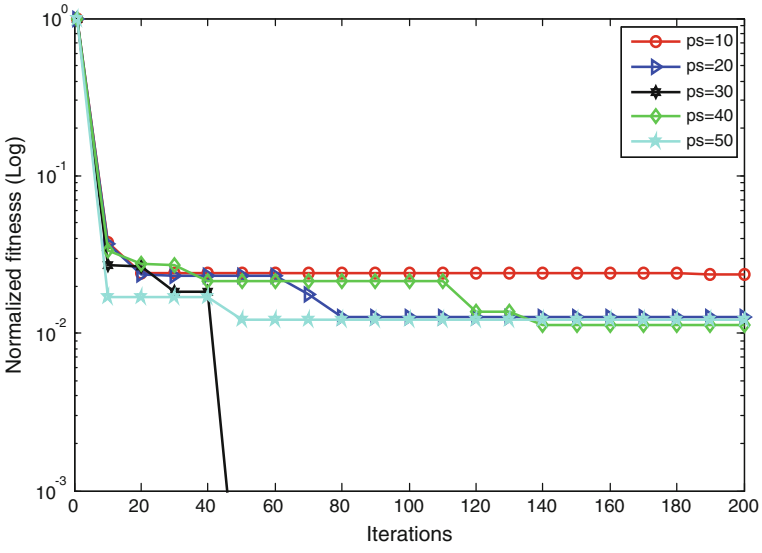

Fig. 9 Convergence curve for *A_n33_k5*

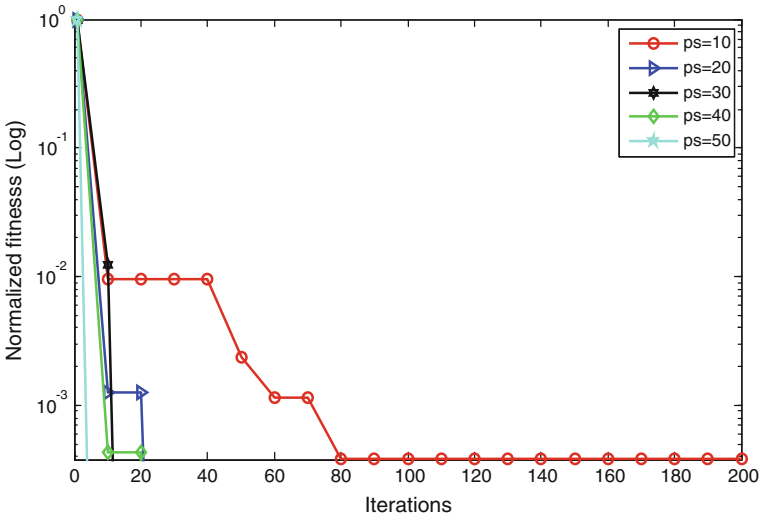

Fig. 10 Convergence curve for *E_n23_k3*

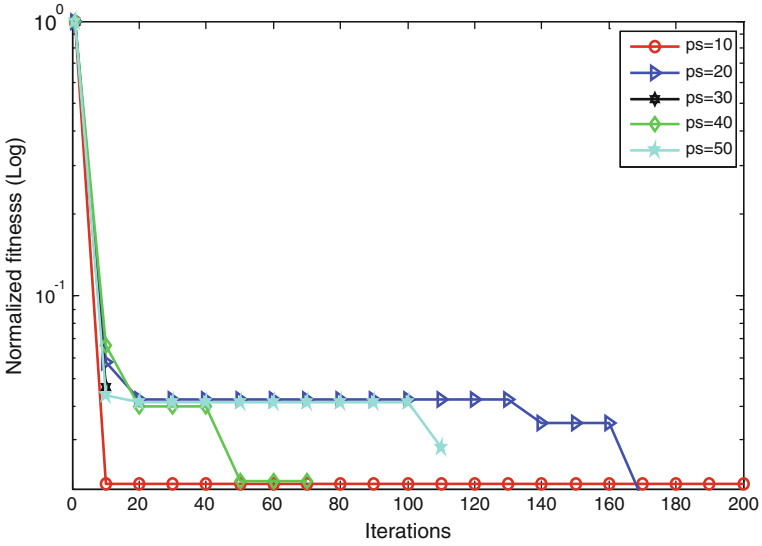

Fig. 11 Convergence curve for *P_n19_k2*

4.2 Comparisons of Simulation Results

In order to show the effectiveness of HBA-PR, we carry out a set of simulations to compare HBA-PR with other state-of-the-art algorithms, i.e., a parallel version of the classical Clarke and Wright Savings (CWS) heuristic, SR-GCWS (Simulation in

Table 1 Comparison of results for Augerat et al. Set A, E, and P instances

Instance	Capacity	Tight-ness	I_BKS	R_BSK	CWS	SR-GCWS	CS-GRASP	HBA-PR
A-n33-k5	100	0.89	661	662.76	712.05	662.11	662.1452	662.1101
A-n33-k6	100	0.9	742	742.83	776.26	742.69	743.5785	742.6933
A-n37-k5	100	0.81	669	672.59	707.26	672.47	672.4652	672.4652
A-n39-k6	100	0.88	831	833.20	863.08	833.20	–	835.2518
E-n23-k3	4500	0.75	569	–	–	–	569.7461	568.5625
E-n22-k4	6000	0.94	375	375.28	388.77	375.28	–	375.2798
E-n33-k4	8000	0.92	839	838.72	843.1	837.67	–	837.9253
E-n51-k5	160	0.97	521	524.94	584.64	524.61	–	524.6111
P-n19-k2	160	0.97	212	212.66	237.90	212.66	212.6569	212.6569
P-n20-k2	160	0.97	216	217.42	234.00	217.42	217.4156	217.4156
P-n22-k2	160	0.96	216	217.85	239.50	217.85	217.8522	217.8522
P-n51-k10	80	0.97	741	742.48	790.97	741.50	–	743.2648

Routing via the Generalized Clarke and Wright Savings heuristic) proposed by Juan et al. [9], CS-GRASP proposed by Zheng et al. [10]. The results of these simulations are summarized in Table 1 which contains the following information of each instance: Vehicle Capacity, Tightness (Demand/Capacity), I_BKS is integral best-known solution (BKS) or 'optimal' value according to the web http://www.branchandcut.org/. R_BSK is verified by the real costs for the best-known solutions according to [9] and CWS is the costs associated with the solution given by the parallel version of the CWS heuristic, while SR-GCWS is the best solution obtain by SR-GCWS method, and HBA-PR is our best solution, where "–" represents no records in the literature.

From the simulation results obtained by testing HBA-PR, it demonstrates that the proposed HBA-PR is effective to solve the CVRP, and the performance of HBA-PR is prominent. From Table 1, all instances achieved a good quality solution, and the solutions of six instances are better than the best-known solutions ('optimal' value). HBA-PR has matched 10 of the 12 best-known solutions except for *P_n51_k10* and *A_n39_k6*, and the average deviation from the real costs best-known solutions is 0.057 %. HBA-PR outperforms CWS for the 12 instances. Compare with CS-GRASP and HBA-PR, HBA-PR outperform CS-GRASP in terms of *A_n33_k5*, *A_n33_k6* and *E_n23_k3,* the other instances have same results. The SR-GCWS and HBA-PR have similar results, and the gap is very small.

Furthermore, Fig. 12 shows the best solution found so far by using our methodology for the *A_n37_k5*.vrp file, where the depot (using 1 instead of 0) is at

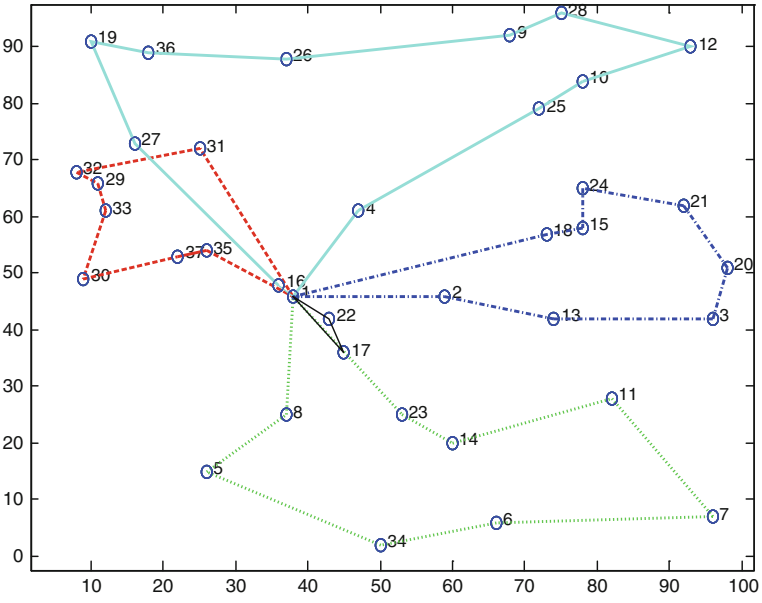

Fig. 12 Optimal routes of *A_n37_k5*

the center. Analogously, Figs. 13, 14 and 15 show the best solutions found by HBA-PR for the *E-n51-k5*.vrp file and *P_n51_k10*.vrp file. The convergence curve for some test problems has been shown in Fig. 14. From Fig. 14, for e.g., *A_n37_k5*, it converges to an optimal solution after 43 generations, and it expends

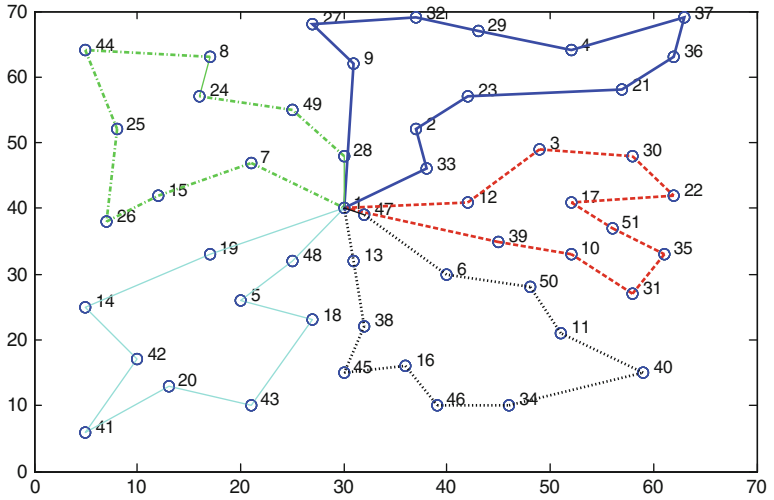

Fig. 13 Optimal routes of *E_n51_k5*

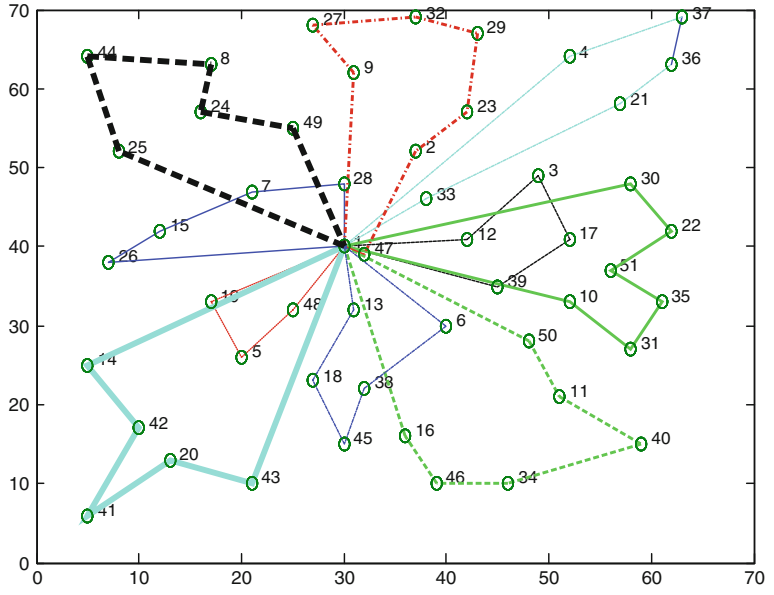

Fig. 14 Optimal routes of *P_n51_k10*

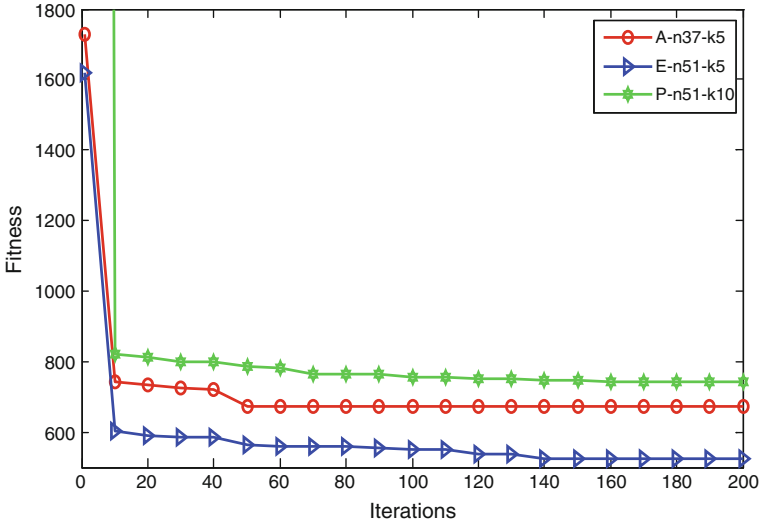

Fig. 15 Convergence curve of three instances

about 170 generations; for e.g., *E-n51-k5* and *P_n51_k10* when the algorithm converges to an optimal solution, which demonstrates that the HBA-PR has a faster convergence rate.

In general, the proposed HBA-PR can produce good solutions when compared with existing heuristics for solving the CVRP, and the convergence rate of HBA-PR is faster. These results seem to indicate that the hybrid bat algorithm with path relinking is an effective alternative to solve the capacitated vehicle routing problem.

5 Conclusions

The capacitated vehicle routing problem is important in the fields of operations research, which is an NP-hard problem. The bat algorithm is a continuous meta-heuristic that cannot be used to solve the CVRP directly. In this paper, a hybrid bat algorithm with path relinking (HBA-PR) for solving the CVRP has been presented. This methodology, which does not require any particular fine-tuning of parameters or configuration process, combines the classical greedy randomized adaptive search procedure (GRASP) with the bat algorithm, and path relinking as an intensification strategy to explore local trajectories connecting elite solutions obtained by proposed algorithm; sub-sequence and single-point local searches are effectively integrated into HBA-PR. The results show that our methodology is able to provide fine-quality solutions which can compete with the ones provided by some exact and heuristic approaches. Moreover, because of its simplicity and flexibility, we believe that this methodology can easily be adapted to solve other variants of the vehicle routing

problem and even to other combinatorial problems, for example, the vehicle routing problem with time windows and the well-known traveling salesman problem, which will form part of our further work.

Acknowledgments This work is supported by the National Science Foundation of China under Grants No.s 61165015 and 61463007. The Key Project of Guangxi Science Foundation under Grant No. 2012GXNSFDA053028, Key Project of Guangxi High School Science Foundation under Grant No. 20121ZD008.

References

1. Dantzig, G., Ramser, J.H.: The truck dispatching problem. Manag. Sci. **6**, 80–91 (1959)
2. Haimovich, M., Rinnooy Kan, A.H.G., Stougie, L.: Analysis of heuristics for vehicle routing problems. In: Golden, B.L., Assad, A.A. (eds.) Vehicle Routing: Methods and Studies. Elsevier, Amsterdam, North-Holland (1988)
3. Toth, P., Tramontani, A.: An integer linear programming local search for capacitated vehicle routing problems. In: The Vehicle Routing Problem: Latest Advances and New Challenges, pp. 275–295. Springer, New York (2008)
4. Nazif, H., Lee, L.S.: Optimised crossover genetic algorithm for capacitated vehicle routing problem. Appl. Math. Model. **36**(5), 2110–2117 (2012)
5. Ai, T.J., Kachitvichyanukul, V.: Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. Comput. Ind. Eng. **56**(1), 380–387 (2009)
6. Szeto, W.Y., Wu, Y.Z., Ho, S.C.: An artificial bee colony algorithm for the capacitated vehicle routing problem. Eur. J. Oper. Res. **215**(1), 126–135 (2011)
7. Chen, P., Huang, H.K., Dong, X.Y.: Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. Expert Syst. Appl. **37**(2), 1620–1627 (2010)
8. Yurtkuran, A., Emel, E.: A new Hybrid Electromagnetism-like Algorithm for capacitated vehicle routing problems. Expert Syst. Appl. **37**(4), 3427–3433 (2010)
9. Juan, A.A., Faulin, J., Ruiz, R., Barrios, B., Caballé, S.: The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem. Appl. Soft Comput. **10**(1), 215–224 (2010)
10. Zheng, H.Q., Zhou, Y.Q., Luo, Q.F.: A hybrid cuckoo search algorithm-GRASP for vehicle routing problem. J. Convergence Inf. Technol. **8**(3) (2013)
11. Yang, X.S.: A new metaheuristic bat-Inspired algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO). SCI **284**, B65–B74 (2010)
12. Gandomi, A.H., Yang, X.S., Alavi, A.H., Talatahari, S.: Bat algorithm for constrained optimization tasks. Neural Comput. Appl. 1–17 (2012)
13. Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. Eng. Comput. **29**, 464–483 (2012)
14. Mishra, S., Shaw, K., Mishra, D.: A new meta-heuristic bat inspired classification approach for microarray data. Procedia Technol. **4**, 802–806 (2012)
15. Xie, J., Zhou, Y.Q., Chen, H.: A novel bat algorithm based on differential operator and Lévy-flights trajectory. Comput. Intell. Neurosci. http://dx.doi.org/10.1155/2013/453812 (2013)
16. Wang, G., Guo, L., Duan, H., Liu, L., Wang, H.: A bat algorithm with mutation for UCAV path planning. Sci. World J. http://dx.doi.org/10.1100/2012/418946 (2012)
17. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. J. Global Optim. **6**(2), 109–133 (1995)
18. Resendel, M.G.C, Ribeiro, C.C.: GRASP with path-relinking: recent advances and applications. In: Metaheuristics: Progress as Real Problem Solvers, pp. 29–63. Springer, New York (2005)

19. Festa, P., Resende, M.: An annotated bibliography of GRASP, part I: algorithms. Int. Trans. Oper. Res. **16**, 1–24 (2009)
20. Festa, P., Resende, M.: An annotated bibliography of GRASP, Part II: Applications. Int. Trans. Oper. Res. **16**, 131–172 (2009)
21. Mestria, M., et al.: GRASP with path relinking for the symmetric euclidean clustered traveling salesman problem. Comput. Oper. Res. http://dx.doi.org/10.1016/j.cor.2012
22. Resende, M.G.C., Ribeiro, C.C., Glover, F., Mart, R.: Scatter search and path-relinking: fundamentals, advances, and applications. In: Gendreau, M., Potvin, J.Y. (eds) Handbook of Metaheuristics, pp. 87–107. Springer, Boston (2010)
23. Glover, F.: Tabu search and adaptive memory programming–advances, applications and challenges. In: Barr, R.S., Helgason, R.V., Kennington, J.L. (eds.) Interfaces in Computer Science and Operations Research, pp. 1–75. Kluwer, Dordrecht (1996)
24. Laguna, M., Martı́, R.: GRASP and path relinking for 2-layer straight line crossing minimization. J. Comput. **11**(1), 44–52 (1999)
25. Bekdaş, G., Nigdeli, S.M., Yang, X.-S.: Metaheuristic optimization for the design of reinforced concrete beams under flexure moments. In: 5th European Conference of Civil Engineering (ECCIE'14), 22–24 Nov 2014, Florence, Italy

Hybrid Metaheuristic Algorithms in Geotechnical Engineering

Y.M. Cheng

Abstract The solutions of many engineering problems can be formulated as the optimized results of a functional. While many engineering problems are governed by a continuous convex optimization process, this is not the case for many geotechnical problems. Many geotechnical problems have irregular solution domains, with the objective function being nonconvex and may not be a continuous function. The presence of multiple local minima is common in many geotechnical problems, and the occurrence of local zones where there is rapid changes in the material parameters is not uncommon. The corresponding governing problems are hence usually NP-type nonconvex optimization problem, and by nature, such NP-type problems with the various constraints pose great difficulty in analysis. While the classical heuristic optimization methods may work well for some of these problems, there are also some practical cases where the classical methods may fail to perform satisfactorily. To maintain a balance between the computation time and accuracy, several hybrid metaheuristic algorithms are proposed by the author which can work well for many practical geotechnical problems. In this chapter, the author will illustrate the basic concept of hybrid metaheuristic algorithms and the applications to some difficult geotechnical problems.

Keywords Hybrid optimization · Geotechnical engineering · Wave equation · Back analysis · Slope stability

1 Introduction

In the solution of many engineering problems, many numerical methods have been developed over the years. The finite element method is the most popular numerical method at present due to its versatility in many complicated problems. The fun-

Y.M. Cheng (✉)
Department of Civil and Environmental Engineering,
Hong Kong Polytechnic University, Hung Hom, Hong Kong
e-mail: ceymchen@polyu.edu.hk

X.-S. Yang et al. (eds.), *Metaheuristics and Optimization in Civil Engineering*,
Modeling and Optimization in Science and Technologies 7,
DOI 10.1007/978-3-319-26245-1_13

damental principle of the finite element is either the variational formulation or the Galerkin formulation [1], which can be viewed as the minimization of energy or functional. The determination of the shortest path, minimum functional or energy is required for many stability problems in geotechnical engineering, as the ultimate condition will provide the factor of safety for use by many engineers. While Denn [2] has demonstrated that the variational principle can be used for optimization analysis, Cheng et al. [3] and Cheng [4] have demonstrated that the optimization process is equivalent to the variational principle and can replace it for use in a variety of problems. Due to the needs for the determination of the optimized solution for various engineering problems, different methods have been proposed with success in different disciplines. Traditionally, resource allocation, packing, and scheduling as well as many other similar problems are analyzed using linear and/or integer programming method. Such methods usually require the objective function and constraints to be linear functions, but the optimum solution can usually be determined rapidly (except under very special cases). Various gradient-type methods have been proposed and used for many engineering problems, and the FSQP (Feasible sequential quadratic programming) and BFGS (Broyden–Fletcher–Goldfarb–Shanno) methods which are variants of the quasi-Newton approach appear to be very effective in many situations with very good performance. Such gradient-type methods require the differentiability of the objective functions (can be formed by finite difference scheme) are, however, limited by the continuity requirement, and the global minimum may not be determined unless a good initial is used in the analysis. In geotechnical engineering and many other disciplines where multiple minima exist in the solution domain, the uses of the previous two groups of methods are seldom adopted. The author has, however, used these types of methods for global optimization in some geotechnical problems by specifying a series of initial starting points, and such strategy appears to be necessary for some large-scale systems where the numbers of variables are in the order of thousands.

In general, most of the practical engineering problems are constrained within acceptable solution domain. The objective functions in many geotechnical or transportation problems are usually NP (nonpolynomial) hard-type problems with the features:

- The solution domain and/or the problem geometry are not regular and smooth. The objective function is commonly nonsmooth and nonconvex, it may not be continuous within the part of the solution domain (failure to converge or infeasible solution).
- Multiple local minima will be present in general, and a good initial trial can be difficult to be specified for general cases. For the one-dimensional function defined over the domain AB as shown in Fig. 1, the global minimum is given by point E, but points C and D may also be obtained if the optimization algorithm cannot escape from the local minimum. If the global minimum is governed by point B where the derivative of the function is not zero, many optimization algorithms (particularly those gradient types) will fail to detect this point.

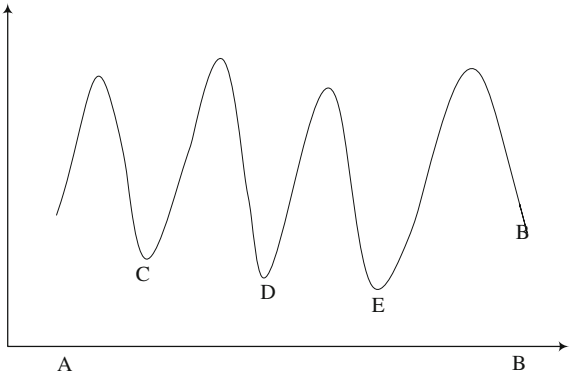

Fig. 1 A simple one-dimensional function illustrating the problem of local minima and global minimum

- The number of control variables is not a small number (usually exceed 20 for slope stability problem, and can exceed thousands for other stability problems) and the solution time for each objective function determination can be demanding. A balance between time and accuracy of solution must be maintained, and the solution algorithm must be robust to suit for various conditions. In particular, the occurrence of local zones where there are rapid changes in the material parameters is not uncommon. Such narrow zones may escape from the optimization search easily, as equal opportunity is usually assigned to everywhere within the solution domain. A simple illustration of such condition is illustrated in Fig. 2 (with the famous Fei Tsui Road slope failure in Hong Kong as a practical case). It is extremely difficult to detect point g for most of the optimization algorithms, as there is a very rapid change of the function (as well as parameters) within a very narrow zone.

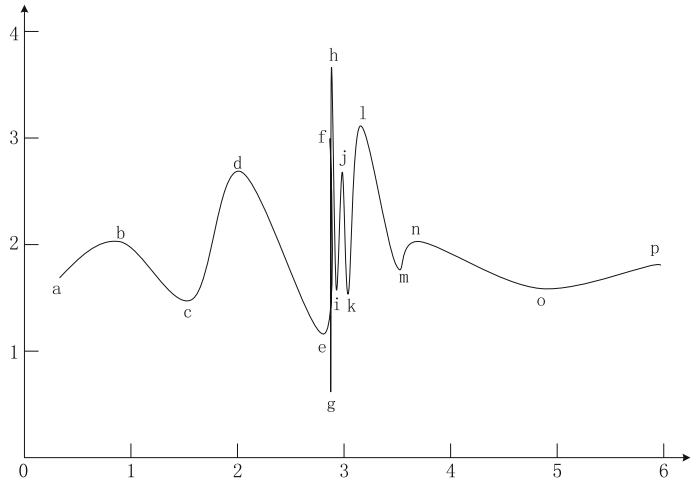

Fig. 2 A complicated one-dimensional function with the presence of several "strong" maxima and minima for illustration

Due to the special requirements of the objective functions in geotechnical engineering, many researchers have adopted different methods to search for the global minimum in geotechnical engineering with various success and limitations. Finite element limit analysis usually adopts those gradient-type methods with the use of different initial solutions in the optimization analysis. Slope stability problem is controlled by limitation 1 as given above, and the gradient-type method is virtually not adopted in practice. For such problems, heuristic optimization methods are more popular at present, and a detailed discussion is given by Cheng [5], Cheng and Li [6], Cheng et al. [7], Cheng et al. [8, 9]. The modern heuristic global optimization methods have attracted the attention of many geotechnical engineers recently due to the simplicity in implementation, satisfactory performance in many practical problems, and increased speed in solution when the number of variables is not excessive. At present, simulated annealing, tabu search, genetic algorithm, particle swarm optimization, fish swarm optimization, harmony search, ant colony method, evolution algorithm, and many other similar variants have been adopted in many geotechnical problems with satisfaction. These methods are suitable for medium-size problems with the presence of multiple local minima. Cheng et al. [7] have carried out a detailed comparison between six major types of heuristic global optimization methods in slope stability problem, and has concluded that no single method can outperform other methods under all cases. That means, every method has its own advantages and limitations, and there is no single method which is universal and can perform well under all cases. A good optimization algorithm should preferably be insensitive to the optimization parameters, except for some very special cases. The author has also carried out the sensitivity analysis of these methods under different optimization parameters which is a very useful and important work but is not commonly considered. It is also observed that some optimization methods (ant colony and tabu search) may be less effective for problem where the objective functions are highly discontinuous, but may be highly efficient when the problem is relatively simple.

2 Example of Geotechnical Problems Which Require Heuristic Optimization Analysis

There are many geotechnical problems which require the use of optimization analysis, in particularly, the pile driving and stability problems. The control of pile driving and the evaluation of the pile capacity are important for pile installation. Currently, there is an increasing trend to adopt the dynamic test instead of the classical kentledge test in order to save the cost of testing as well as examining more piles. The pile driving wave equation for a uniform pile is given by:

$$\frac{\partial^2 u}{\partial t^2} = C^2 \frac{\partial^2 u}{\partial x^2} + R \tag{1}$$

where u is the axial displacement which is a function of both the depth x and time t, C is the speed of stress wave, R is the skin resistance of soil around the pile which is a function of x and t. Based on the theory of characteristics, the integral along the positive and negative directions of the characteristics line gives the following iterative form:

$$\text{Along}\frac{\mathrm{d}x}{\mathrm{d}t} = C, \quad Z_{i-1}V_{i,j} + F_{i,j} = Z_{i-1}V_{i-1j} + F_{i-1j} - R_{i-1}^{+} \tag{2}$$

$$\text{Along}\frac{\mathrm{d}x}{\mathrm{d}t} = -C, \quad Z_iV_{i,j} - F_{i,j} = Z_iV_{i+1,j-1} + F_{i+1,j-1} - R_i^{-} \tag{3}$$

$$\text{where } R_{i-1}^{+} = \frac{R_u(i-1)}{O_k(i-1)}\left(u_{i-1j} - u_{pi-1j}\right)\left(1 + J_{si-1}V_{i-1,j}\right) \tag{4}$$

$$R_i^{-} = \frac{R_u(i)}{Q_k(i)}\left(u_{i+1,j-1} - u_{pi+1,j-1}\right)\left(1 + J_{si}V_{i+1,j-1}\right) \tag{5}$$

Z, V, F, Q, R_u, Q, and J are the impedance, velocity of wave, force in pile, elastic limit of skin friction, ultimate skin resistance, quake, and damping constant of the Smith model, respectively (see [10] for details). After a time interval Δt, upward $P_u(i-1,j)$ and downward traveling waves $P_d(i-1,j-1)$ become $P_{(i,j)}$ at the top of the unit. When $t = j\Delta t$, considering the transmission and reflection of the upward and downward traveling waves, and the upward and downward traveling waves of the skin resistance $R(i,t)$, the upward and downward traveling waves at section i are, respectively, given as follows:

$$P_u(i,j) = 2\cdot\frac{Z_i}{Z_{i+1}+Z_i}P_u(i+1,j) + \frac{Z_{i+1}-Z_i}{Z_{i+1}+Z_i}P_d(i,j-1) + \frac{Z_i}{Z_{i-1}+Z_i}R(i,j) \tag{6}$$

$$P_d(i,j) = 2\cdot\frac{Z_i}{Z_i+Z_{i-1}}P_u(i-1,j-1) + \frac{Z_{i-1}-Z_i}{Z_i+Z_{i-1}}P_u(i,j-1) - \frac{Z_{i+1}}{Z_i+Z_{i+1}}R(i,j) \tag{7}$$

At the pile tip, the upward and downward traveling waves will be:

$$P_d\left(N_{p,j}\right) = P_d\left(N_{p-1,j-1}\right) \tag{8}$$

$$P_u\left(N_{p,j}\right) = -P_d\left(N_{p-1,j-1}\right) + R\left(N_{s,j}\right) + R\left(N_{s+1,j}\right) \tag{9}$$

At time t, the measured force and speed near to the top of pile are $P_m(j)$ and $V_m(j)$. If we take the location of the PDA sensors as the boundary and the measure velocity as the boundary condition to compute the force in pile, we have

$$P_u(1,j) = P_u(2,j-1) \tag{10}$$

$$P_d(1,j) = ZV_m(j) + P_u(2,j-1) \tag{11}$$

Therefore, the force–time curve can be obtained by:

$$P_c(j) = P_d(i,j) + P_u(i,j) = ZV_m(j) + 2P_u(2,j-1) \tag{12}$$

The particle velocity and displacement $V(i,j)$ and $S(i,j)$ are given by:

$$V(i,j) = \frac{P_d(i,j)}{Z_i} - \frac{P_u(i,j)}{Z_{i+1}} \tag{13}$$

$$S(i,j) = S(i,j-1) + \frac{\Delta t}{2}[V(i,j-1) + V(i,j)] \tag{14}$$

If the Smith soil model is used, three parameters will be defined: the maximum static resistance R_u, the largest elastic deformation Q, and the Smith damping coefficient J_s. Soil resistance $R(i,j)$ is divided into static resistance $R_s(i,j)$ and dynamic resistance $R_d(i,j)$ as given by

$$R(i) = R_s(i,j) + R_d(i,j) \quad \text{and} \quad R_d(i,j) = |R_s(i,j)|J_s(i)V(i,j) \tag{15}$$

According to the above-mentioned formulas, force–wave curve $F_c(t)$ can be calculated according to the measured $V_{m(t)}$ for different time and segments by successive cycling of the above processes.

The force in pile based on the calculation is determined at different time step. For matching of the signal in order to determine the pile capacity and soil parameters, the difference between the calculated and the measured force values can be represented by an objective function *F(x)* as Eq. (16)

$$F(x) = \sum_{j=1}^{N_{\text{time}}} \text{ABS}(F_c(j) - F_m(j)) \tag{16}$$

where $x = \{(R_u(i), Q(i), J_s(i), R_t, Q_t, ff, W_s, J_{\text{ms}}), i = 1, N_{\text{pile}}\}$ are the unknowns vector and N_{time} is the number of time–step (usually taken as 1024). The best solution will be given by:

$$\text{Minimize}\, F(x) = \sum_{j=1}^{N_{\text{time}}} \text{ABS}(F_c(j) - F_m(j)) \tag{17}$$

Based on the critical results, the pile capacity and soil parameters for the dynamic pile test can be determined which is only 5–10 % the cost for the traditional

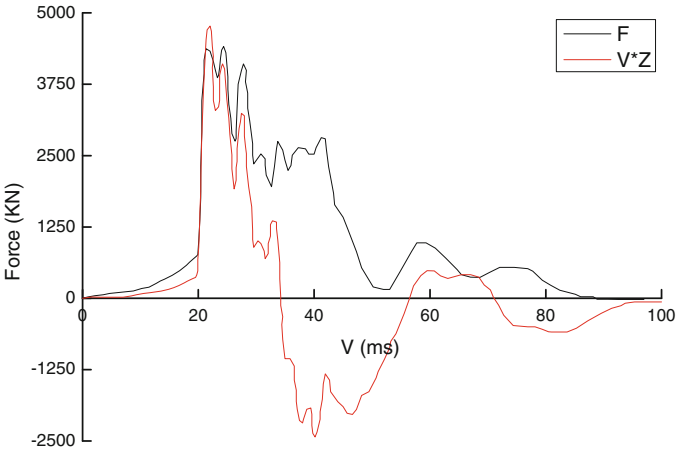

Fig. 3 A PDA test result for pile driving in sandy soil in Hong Kong

kentledge test. For large-strain PDA test, the force and velocity during pile driving are measured, and a typical PDA test result in Hong Kong is shown in Fig. 3.

In the PDA backward analysis, the skin friction, soil parameters, and pile capacity become the unknowns. It is possible to vary these parameters until the calculated signal can match well with the measured signal. If this condition is achieved, the pile capacity as well as the skin friction and soil parameters will be obtained. This problem is well known to have many local minima during the analysis, and great care and experience are required for the proper analysis. Many engineers perform the analysis in two ways: (1) prescribed the quake and damping constants (usually constant throughout the pile) and manually adjust the skin resistance and end bearing capacity; (2) perform an automatic matching analysis based on some optimization algorithm in computer program. Manual trial and error by approach 1 by program CAPWAP (Rausche et al. [11]) is time consuming and relies heavily on engineer's experience, but it can avoid unreasonable results which may come out from automatic signal matching. On the other hand, automatic signal matching is fast in operation, but unreasonable soil parameters which can give good signal matching cannot be avoided in the analysis. In fact, the author has observed unreasonable output results from the optimization analysis prepared by the engineers based on unconstrained automatic optimization analysis (no restraint on the ranges of the variables), and such problem can partly be avoided by the use of constrained optimization search. By nature of this pile driving problem, there are many local minima in the solutions, and it is not easy to search for the global minimum.

Besides the pile driving problem, the slope stability problem using limit equilibrium or finite element analysis as well as the finite element limit analysis are typical problems which require the use of various types of optimization method.

3 Hybrid Heuristic Optimization Methods

In view of the limitations of the classical optimization methods, the current approach to locate the critical failure surface for slope stability analysis is by the heuristic global optimization methods. The term heuristic is used for algorithms which find solutions among all the possible ones, but they do not guarantee that the best will be found; therefore, they may be considered as approximate and not accurate algorithms. These algorithms usually find a solution close to the best one, and they find it fast and easily. The early applications of such method in geotechnical engineering in slope stability are due to Greco [12] and Malawi et al. [13] using the Monte Carlo technique for locating the critical slip surface with success for some cases, but there is no precision control on the accuracy of the global minimum. Zolfaghari et al. [14] adopted the genetic algorithm while Cheng [5, 15], Cheng and Li [16], and Cheng et al. [7–9, 17] have adopted and improved many heuristic optimization algorithms for use in slope stability problems. The author has also developed a slope stability program SLOPE 2000 which is widely adopted for use in various countries (Hong Kong, China, Taiwan, Europe) as well as the slope stability module in a large geotechnical analysis package GEOSUITE 2.0 (mainly used in Europe). In these two programs, the optimization algorithms can be chosen by the users, and the algorithms include: simulated annealing (SA), ant colony (ANT), particle swarm, and modified particle swarm (PSO), harmony search and modified harmony search (HM), fish swarm search (FS), genetic algorithm (GA) and tabu search. These basic optimization algorithms are discussed by Cheng and Lau [18] in details and will not be repeated here. The author has also proposed the concept of weighted random number [5] for the case of a narrow band where the material parameters can change significantly within a narrow region. Based on the actual application of these improved heuristic optimization algorithms over varieties of problems in different countries, the optimization algorithms in the two programs are now very stable and robust and can pass through very difficult problems with only very few exceptions. A demonstration version of the program can be downloaded from the author's website at http://www.cse.polyu.edu.hk/~ceymcheng/.

Heuristic algorithms are usually approximate and are not accurate algorithms. These algorithms usually find a solution close to the best one effectively and efficiently. Every heuristic algorithm relies on the use of some parameters for analysis, but there is no rigorous method in determining these kinds of parameters for general case (not to mention those where there is a sudden change of material parameters in the solution domain). The success and efficiency of a global optimization algorithm may rely on the use of these parameters. Cheng et al. [7] have found that no single method can outperform other methods under all cases, and every method can fail to work under some cases (not common). The performance of a good optimization method should be relatively insensitive to the optimization parameters. Every global optimization method can be tuned to work well if suitable optimization parameters are adopted, but such parameters are difficult to be

established for a general problem. For sake of safety, the author allows the users to choose different optimization algorithms in the analysis in the above two programs, in case the users have doubt on the acceptability of the optimized solution.

The author has come across some complicated hydropower projects in China where there are several 'strong' local minima in the solution domain (similar to the case in Fig. 2). The engineers have used different commercial programs with different results, and there is a lack of confidence on the results of analysis. To deal with these cases, the author has proposed a coupled optimization procedure based on the PSO and HS methods for this special problem. The coupled method is later extended to other optimization algorithms to take advantage of different optimization algorithms. In general, the hybrid (coupled) optimization method is more stable and robust than the original optimization methods, at the expense of longer computation time. In fact, the author views that there is no simple way to maintain effectiveness and efficiency at the same time in general. Since the increase in the computation is usually not significant, the author has also put the hybrid optimization algorithm into the two programs as mentioned previously. In the following sections, some hybrid optimization will be discussed, which will then be illustrated by examples for assessing the performance of such hybrid methods.

4 Hybrid Particle Swarm Harmony Search Optimization

In the PSO method, the positions of the particles are updated by modifying the corresponding velocity vectors. If the choice of ω is not appropriate, it may lead to the trap into the local minimum, where ω is the inertia weight coefficient. In general, a value of 0.5 for ω is used normally which is found to be adequate for most of slope stability problem. Alternatively, a larger value of ω can be applied at the initial search, which is then gradually reduced to a smaller value for refined results near to the existing best position. Another approach is suggested by Wang and Liu [19] as shown in Eq. (1) which is adopted in the present study: the current positions of particles, the best position found so far P_i, and the best position of any particle within the context of the topological neighborhood of the ith particle found so far P_g.

$$\begin{aligned} V_i^{k+1} &= \omega V_i^k + c_1 r_1 \left(P_i - X_i^k\right) + c_2 r_2 \left(P_g - X_i^k\right) \\ X_i^{k+1} &= X_i^k + V_i^{k+1} \quad , i = 1, 2, \ldots, 2n \end{aligned} \tag{18}$$

where X_i, V_i, P_i, and P_g are the position, velocity, the difference between the ith particle's best position found so far and the current position, and the difference between the best position of any particle within the context of the topological neighborhood of the ith particle found so far. c_1 and c_2 are the stochastic weighting which are chosen to be 2 while r_1 and r_2 are two random numbers in the range [0,1]. A larger value for ω will enable the algorithm to explore the search space, while a

smaller value of ω will lead the algorithm to exploit the refinement of the results. The HS method is another efficient and effective global optimization method for many geotechnical problems which is discussed by Cheng et al. [7]. Cheng and Lau [20] have also given a detailed procedure in the implementation of the modified harmony search algorithm which is adopted in the present hybrid scheme. When the problem size is large, HS can be trapped by the local minimum easily. Cheng et al. [9] have proposed a modified harmony (MHS) search method to overcome the limitation of the original harmony search method. The utilization of the MHS is by generating several new harmonies than by generating a new harmony during each iteration. Two parameters *HR* and *PR* for harmony search are required in the analysis, and the detailed procedure is shown in Fig. 4.

If we take the above-mentioned positions (flights) from PSO as the harmonies in the Harmony Memory in HS, a new position can also be obtained by the harmony search procedure. In Fig. 4, $z_{i\min}$ and $z_{i\max}$ are the minimum and maximum values of the ith element in vector X. z_{ij} is the jth element of X_i. Similar to the modified PSO, N_a ($\leq M$) flights within each iteration step are allowed with different approach. It is possible to choose N_a particles randomly from the total generation rather than based on the fitness of the particles in the modified PSO. This is a minor and simple trick in combining the two methods. Cheng et al. [7] have tried genetic algorithm, simulated annealing method, PSO, HM, tabu search, and ant colony search, and have commented no single method can outperform other method under all cases. Each optimization method has its own merits and limitations, and the combination of two optimization methods can possibly result in a better performance under difficult cases which will be illustrated.

The flowchart for the hybrid PSO and HS which is denoted as HS/PSO is shown in Fig. 5. It should be noted that the flowchart in Fig. 5 is a simple combination of the PSO and HS methods, and the author do not attempt to propose a highly complicated procedure in combining these two methods for sake of simplicity and ease of

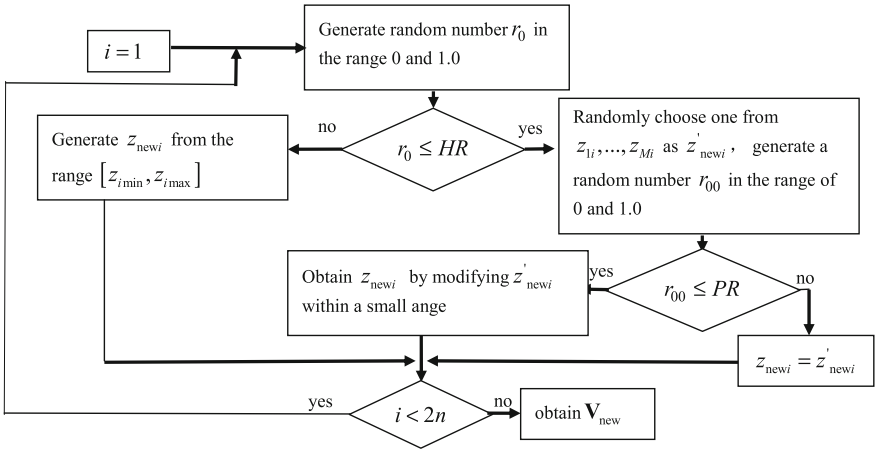

Fig. 4 Generation of a new harmony and the search procedure

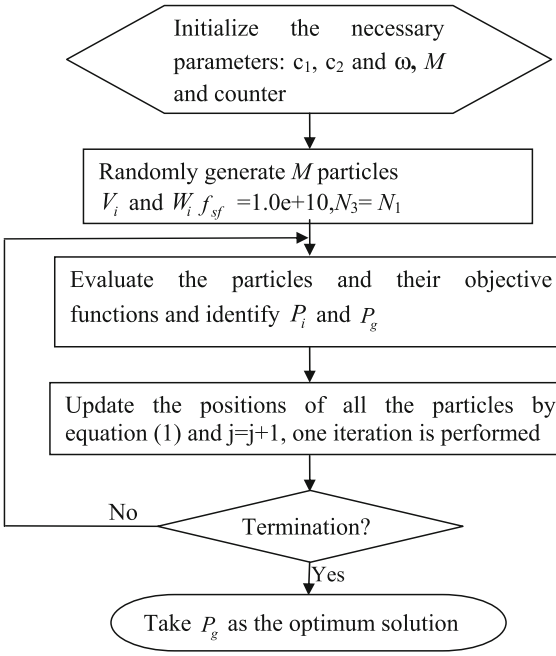

Fig. 5 The flowchart for the hybrid HS/PSO Optimization method

implementation. The step on the updating of the positions of all the particles in the PSO method is replaced by the harmony search generation as given in Fig. 4. Such a minor change can retain the simplicity of both optimization methods so that the proposed algorithm is simple to use and does not require major computer memory. The author has come across several very complicated cases in some projects, and the presently proposed algorithm combine two optimization methods so that the coupled algorithm will be more stable and robust for very complicated problem. It is true that the present method will be less efficient for simple method and it is not recommended for such purposes even though the hybrid method is still effective for such cases. The proposed algorithm is targeted toward complicated problems (discontinuous objective function with multiple strong local minima and sudden major change in the material properties) for which the other algorithms may fail to perform satisfactorily.

Besides the coupling of PSO and HM, it is also possible to couple the tabu search, simulated annealing, genetic algorithm with the harmony search method, and the author has also successfully implemented these coupling methods.

5 Hybrid Chaos Harmony Search Algorithm (CH/HS)

The chaos phenomenon is first discovered from the simulation of atmospheric turbulence between two infinite planes by the American meteorologist Lorenz in 1963. The solution of the equation is not only stochastic but also nonperiodic if

suitable parameters are selected, which means that a determinate equation could yield stochastic results. Therefore, it is a kind of behavior between random and rules. However, chaos contains refined internal structure instead of being intricate and disordered, which can confine the systematic motion to a specific scope. Chaotic motion shows strong randomicity, ergodicity, and regularity, and it can change a lot even if there is only a small change to the initial conditions. Based on the ergodicity of chaos variables, it is feasible to explore the solution space.

In order to explore the solution space by using chaos variables, the initial values should be obtained first and then the common Logistic Map is adopted to update them. Generally, the steps for exploring the chaos are as follows:

- If $k = 0$, then the initial chaos variable $\text{chaos}^k = \left(\text{ch}_1^k, \text{ch}_2^k, \ldots, \text{ch}_m^k\right)$, where m is the number of designed variable. At the same time, the bounds to the variables are defined, i.e., $U = (u_1, u_2, \ldots, u_m), L = (l_1, l_2, \ldots, l_m)$, where u_j, l_j are the upper and lower bounds of the control variable, respectively.
- According to Eq. (19), the chaos variables chaos^k can be projected as a point in the variable space $R^k = \left(r_1^k, r_2^k, \ldots, r_m^k\right)$ as;

$$r_j^k = l_j^k + \left(u_j^k - l_j^k\right) \times \text{ch}_j^k, j = 1, 2, \ldots, m \tag{19}$$

- The initial values of the chaos variables are then updated using Eq. (20) before turning back to step (2) for continuous iteration, which can find a series of solutions, i.e., $R^0, R^1, \ldots$, and from which the best one can be considered as the solution of optimization problem

$$\text{ch}_j^{k+1} = \text{ch}_j^k \times \left(1 - \text{ch}_j^k\right) \times 4.0, j = 1, 2, \ldots, m \tag{20}$$

It should be noted that the chaos variables have strong ergodicity and can explore solution space from the preceding steps, but there are also such limitations as bad application of the optimal solution and poor development capability of the algorithm. However, new algorithm with strong global search ability can be developed based on the combination of harmony algorithm and advanced exploring ability of chaos algorithm. The author proposes three chaos exploration strategies for research, namely simple chaos exploration strategy (SCHM), static partitioning chaos exploration strategy (SPCHM), and dynamic partitioning chaos exploration strategy (DPCHM).

- Simple chaos exploration strategy is to keep the variable value interval constant during the exploration, which is given in Eq. (19), that is,

$$u_j^k = u_j; \quad l_j^k = l_j, j = 1, 2, \ldots, m. \tag{21}$$

- Static partitioning chaos exploration strategy is to divide the user's given variable value interval into several subintervals, and with more subintervals, the

more computational works will be required. Herein, three intervals are selected as follows:

$$\begin{aligned} U_1 &= (u_1, u_2, \ldots, u_m), L_1 = (f_1, f_2, \ldots, f_m) \\ U_2 &= (f_1, f_2, \ldots, f_m), L_2 = (d_1, d_2, \ldots, d_m) \\ U_3 &= (d_1, d_2, \ldots, d_m), L_3 = (l_1, l_2, \ldots, l_m) \end{aligned} \tag{22}$$

where $f_j = u_j - \frac{u_j - l_j}{3}, d_j = u_j - \frac{u_j - l_j}{3} \times 2, j = 1, 2 \ldots m$. During chaos exploration, three different initial chaos values $\text{chaos}^{k,1}, \text{chaos}^{k,2}, \text{chaos}^{k,3}$ are formed and then projected to a series of solutions in $U_1, L_1; U_2, L_2; U_3, L_3$, respectively. As for $\text{chaos}^{k,1}$, it has been reflected in Eq. (19) that $u_j^k = u_j; l_j^k = f_j, j = 1, 2, \ldots, m$. The best solution among the three series is then chosen as the solution of the optimal problem.

- Dynamic partitioning chaos exploration strategy, namely, solution space is divided into three subintervals utilizing the worst point in the current harmony library $R_b = (r_{b1}, r_{b2}, \ldots, r_{bm})$, and the center point of the other remained points

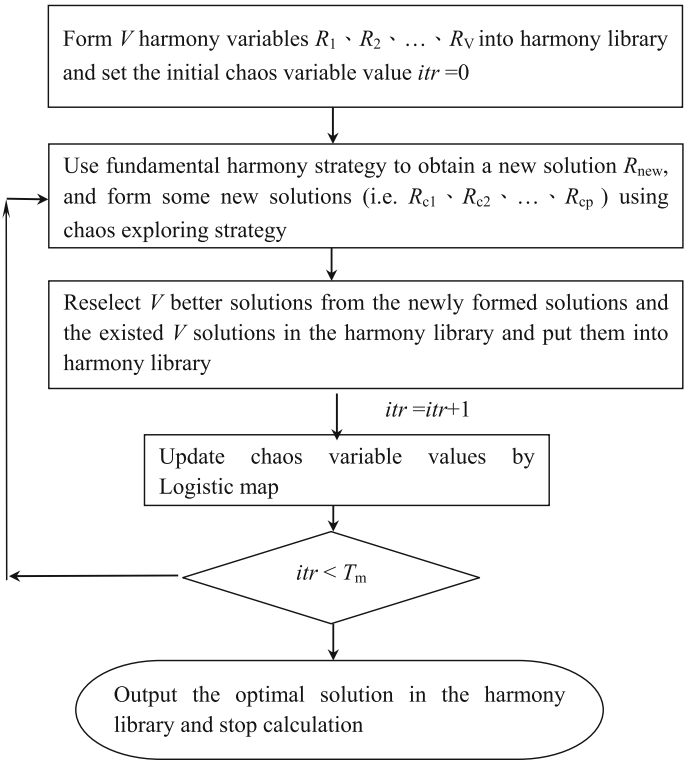

Fig. 6 The flowchart of chaos harmony search algorithm

$R_o = (r_{o1}, r_{o,2}, \ldots, r_{om})$. The definition of the interval is analogous to those depicted above except for small differences in the values of f_j, d_j. Herein, f_j, d_j are the maximum and minimum among the worst point and the center point, respectively. i.e., $f_j = \max(r_{bj}, r_{oj}); d_j = \min(r_{bj}, r_{oj})$.

The flowchart for the hybrid chaos harmony search method is shown in Fig. 6, in which T_m denotes the user's defined maximum number of iteration. Different chaos exploring strategies will give rise to different Chaos Harmony Method. In the process of finding local factors of safety, maximization and minimization are simulated by the fundamental harmony method and Chaos Harmony Method, respectively.

6 Hybrid Genetic Algorithm with Harmony Search Strategy (GA/HS)

It is also possible to couple the genetic algorithm, harmony search, simulated annealing algorithm, tabu search and other methods easily by simple fine tuning in the generation of new trials. In this section, another hybrid method will be discussed, but the author will not try to discuss any more hybrid methods in this chapter. The main reason is that most of the hybrid methods can perform better than the simple methods, but again no single hybrid method seems to outperform other hybrid methods under all cases, in terms of effectiveness and efficiency. Before the discussion on the hybrid genetic harmony search, the two basic methods are explained in more details before the hybrid method can be discussed.

6.1 Harmony Search Algorithm

In music playing, musicians repeatedly adjust the tone of the instrument in the band (based on memory) so as to eventually reach a wonderful state of harmony. The Harmony Search Algorithm is proposed by Geem based on the inspiration of considering instrument i (i = 1, 2, …, m) and harmony of the instrument tone R_j (j = 1,2, … s) analogous to the ith design variable and jth solution vector in the optimization problem. It has been successfully adopted in the combinational optimization problems such as slope stability problem, TSP, pipe laying and public transport line, experimental parameter estimation and others. In this algorithm, M initial solutions (i.e., harmony) are obtained first and put into the Harmony Memory. It will then search new solutions in HM with a probability of HR while searching outside HM with a probability of 1-HR. Thereafter, local disturbance will be added to the new solutions with a probability of PR. If the objective function value of the new solution is better than the worst solution in HM, then replace it. Iteration will not stop until reaching the maximum evolution generation $T_{\max}$ and

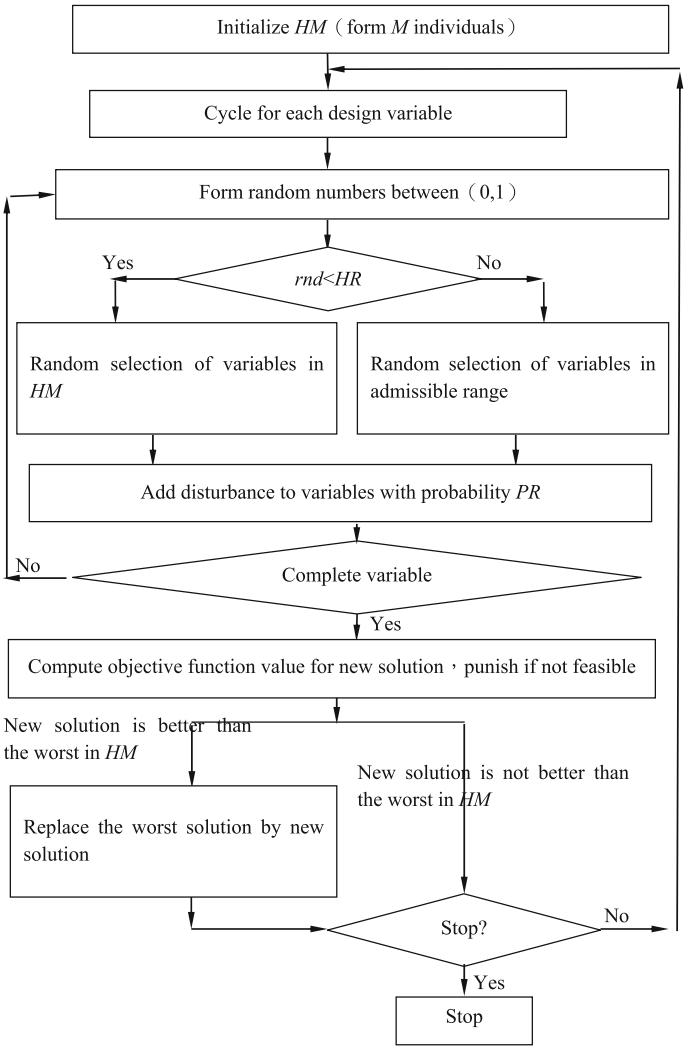

Fig. 7 The flowchart of fundamental harmony search algorithm

the computational process is shown in Fig. 7. Fundamental harmony algorithm used in this chapter adopts punishment and repair strategies to process nonfeasible solutions. From Fig. 7, it can be seen that harmony search algorithm belongs to single individual evolution algorithm. Although the harmony library is used to yield new solutions, only one solution can be found in each evolution. As the approach to form new solutions is very novel, it can be applied in other group evolution algorithms, for example, genetic algorithm, to enhance the probability to find new solutions during the search, and such newly combined algorithm will be of higher global search ability.

6.2 Genetic Algorithm

Genetic algorithm is an adaptive global optimization algorithm based on Darwin's theory of evolution and Mendel's genetic theory, which was first proposed by J.H. Holland. It is mainly implemented by three operators: selection, hybridization, and mutation. Taking the critical slip surface search of soil slope as an example, the principal procedures of genetic algorithm are as follows:

- Determine the design vector of the optimization problem, for example, $X = (x_A, x_B, \ldots, y_{n-1}) = (z_1, z_2, \ldots, z_{n+1})$ in Eq. (23), and objective function value G controlled by design vector (i.e., chromosome) $\boldsymbol{X}$. Regarding to the optimization problem of finding the minimum value of objective function, if G is smaller, then corresponding to which the chromosome $X_i = \left(z_1^i, z_2^i, \ldots, z_{n+1}^i\right)$(where z_j^i denotes the jth gene of the ith chromosome $j = 1, 2, \ldots, n+1$) will be a better adaptation. Then N chromosomes $X_1, X_2, \ldots, X_N$ satisfying constrained conditions are formed randomly and put into the match pool with crossover probability p_c and mutation probability p_m. Moreover, evolution generation $t = 0$ and $T_{\max}$ denotes the maximum evolution generation.

$$\begin{cases} \text{Min} \quad G(x_A, x_B, y_1, y_2, .., y_{n-1}) \\ X_{\text{LA}} \le x_A \le X_{\text{UA}} \,;\; X_{LB} \le x_B \le X_{\text{UB}};\; Y_{\text{L}i} \le y_i \le Y_{\text{U}i} \,,\; i = 1, 2, .., n-1 \end{cases} \tag{23}$$

- $N/2$ pairs of parents chromosomes are formed by randomly selecting two chromosomes out of N in the match pool. For each parent chromosomes, crossover operation can be implemented according to p_c. If yes, two offsprings S_1, S_2 will be obtained by operating arithmetic crossover operator on the parent chromosomes, and then S_1, S_2 are put into children's pool.
- Mutation should be operated on each offspring in the children's pool according to p_m. If yes, the existed offspring should be substituted by another new offspring formed by the nonuniform mutation operator.
- Compute the objective function value of N chromosomes in the match pool and K chromosomes in the children's pool, respectively. Punishment and repair strategies should be adopted if the chromosome is not feasible.
- The objective function values of $N + K$ chromosomes are considered to be in an ascending sort order $G_{m_1}, G_{m_2}, \ldots, G_{m_{N+K}}$. The probability of choosing m_i chromosome is $\rho_{m_i} = \alpha(1-\alpha)^{i-1}, \alpha \in [0 \quad 1]$. N chromosomes are reselected and put into the match pool according to the probability of being chosen for each chromosome. Specifically, those chromosomes with smaller objective function values will be much more than the others in the match pool.
- $t = t + 1$, if $t < T_{\max}$, then turn back to step (2), otherwise, output the optimal chromosome and stop iteration.

From the above, it is clear that the match pool in the fundamental genetic algorithm will be occupied quickly by several excellent chromosomes if the selection pressure is extremely high, that is, α is too large. Generally, it is quite difficult to generate offspring through crossover as well as mutation, because the mutation rate is usually very small. On the basis of the approach prompting new solutions in harmony search algorithm, the harmony genetic algorithm is thus proposed.

7 Genetic Algorithm with Harmony Strategy

The match pool in the genetic algorithm is considered as the harmony library *HM*, and it will produce new solutions if probability *HR* and *PR* are known. The iteration procedures of harmony genetic algorithm are as follows:

- Under the given probability *HR* and *PR*, $N = 2 \times (n+1)$ initial trial $X_1, X_2, \ldots, X_N$ satisfying the constrained conditions are formed randomly and put into the match pool with crossover probability p_c and mutation probability p_m. At evolution generation $t = 0$, $T_{\max}$ and P_{hm} denote the maximum evolution generation and harmony probability, respectively.
- $N/2$ pairs of parents chromosomes are formed by randomly selecting two chromosomes out of N in the match pool. For each parent chromosomes, crossover operation can be implemented according to p_c. If yes, two offsprings S_1, S_2 will be obtained by operating arithmetic crossover operator on the parent chromosomes, and then S_1, S_2 are put into children's pool.
- Mutation should be operated on each offspring in the children's pool according to p_m. If yes, the existing offspring should be substituted by another new offspring formed by the nonuniform mutation operator.
- According to the N chromosomes in the match pool, $N \times P_{\text{hm}}$ new solutions $H_1, \ldots, H_{N \times P_{\text{hm}}}$ are produced utilizing harmony algorithm, and then they are put into the children's pool.
- Compute the objective function value of N chromosomes in the match pool and K chromosomes in the children's pool, respectively. Punishment and repair strategies should be adopted if the chromosome is not feasible.
- The objective function values of $N + K$ chromosomes are considered to be in an ascending order $G_{m_1}, G_{m_2}, \ldots, G_{m_{N+K}}$. The probability of choosing m_i chromosome is $\rho_{m_i} = \alpha(1-\alpha)^{i-1}$, $\alpha \in [0 \quad 1]$. Currently, N chromosomes are reselected and put into the match pool according to the probability of being chosen of each chromosome. Specifically, those chromosomes with smaller objective function value will be much more than the others in the match pool.
- $t = t + 1$, if $t < T_{\max}$, then go back to step (2), otherwise, output the optimal chromosome and stop iteration.

8 Application of Hybrid Optimization Method

For the large-strain pile driving PDA test results in Fig. 3, the length and diameter of the precast pretensioned concrete pile are 26 m and 450 mm, respectively. The Young's modulus of the concrete pile is back-calculated as 45,250 MPa, based on the time for the reflected signal to travel to the pile top. From this figure, it is directly found that the maximum impact force from the diesel hammer is 4787 kN. Two local peaks are observed besides the first peak which is the measured force due to direct striking by the hammer. These two local peaks represent the force reflected from the junctions between pile segments at a depth of 12 and 24 m below pile top. Based on the minimization of Eq. (17) using commercial program, the end bearing, skin friction, and static pile capacity are found to be 4103, 1942, and 5853 kN, respectively, with a prescribed damping factor of 0.1 s/m and a quake of 2.5 mm using signal matching approach 1 (commonly adopted parameters using manual trial and error). If completely automatic signal matching method is used (signal matching approach 2 with unconstrained bounds), an even better signal matching is obtained with a damping factor of 0.5 s/m and a quake of 1.3 mm (which are outside the normal ranges in Hong Kong), and the static pile capacity is estimated to be 5415 kN with some major fluctuation in the skin friction at the middle of the pile. Multiple solutions in signal matching are commonly encountered by many engineers, and the major differences between soil parameters and pile capacities obtained from different set of matching are not uncommon. The static pile capacity by the CASE analysis (Case damping factor = 0.15) is 5765 kN while the pile capacity is found to be 6231 kN from static load test. Based on the coupled optimization HS/PSO as discussed in this chapter, Eq. (17) is solved by assigning an upper and lower bounds to each variable (quake, damping, soil parameters, skin resistance, and base capacity) based on experience of the engineers, and a static pile capacity of 5946 kN (different parameters to different segments of pile) is obtained. Since the upper and lower bounds are established by soil mechanics principle and experience, unrealistic values can be avoided in the signal matching process. This approach possesses the advantages of manual control as well as automatic signal matching, and has been found to be efficient based on some projects in Hong Kong. Using CH/HS and GA/HS, the optimized results are virtually the same as that by the HS/PSO. Since there is no sharp change in the soil parameters, the static pile capacity of 5946 kN can be accepted with confidence, even though this problem has many local minima in the solution domain.

For slope stability problem, a problem with a soft band which has been considered by Zolfaghari et al. [14], Cheng et al. [9] and Kahatadeniya et al. [21] is shown in Fig. 8. The soil parameters for this problem are shown in Table 1. It is noticed that the soil parameters for soil layer 3 are much lower than the other three soils so that the failure surface will be mainly controlled by this layer of soil at the middle of the failure surface. For minimization of the factor of safety, the minimum value using Spencer method for this problem are 1.50, 1.11, 1.361, and 1.09 by the genetic algorithm [14], the artificial fish swarm algorithm (AFSA) [9], the ant

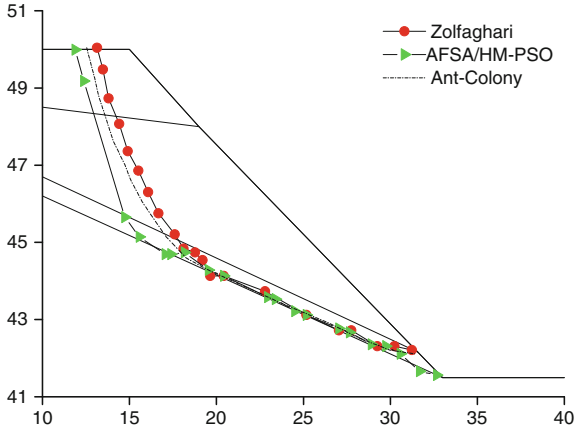

Fig. 8 A slope stability problem with a soft band

Table 1 Geotechnical parameters for the soils in Fig. 8 (γ = unit weight of soil, c′ = effective cohesive strength and ϕ' = effective angle of friction)

Layers	γ(kN/m^3)	c' (kPa)	φ' (°)
1	19.0	15.0	20.0
2	19.0	17.0	21.0
3	19.0	5.00	10.0
4	19.0	35.0	28.0

colony method [21] and HS/PSO. Again, the result by CH/HS and GA/HS are virtually the same as that by the HS/PSO, which indicate that the performance of these three hybrid optimization method are nearly the same for this problems. Since the soft band soil is a strong local minimum but the thickness of this layer is relatively small, the genetic algorithm and the ant colony method cannot escape from the local minimum and fail to provide a good solution for this problem. On the other hand, the artificial fish swarm algorithm and the three hybrid methods in this chapter provide solutions which are nearly the same. In fact, it is difficult to differentiate precisely the critical slip surfaces from the AFSA, HS/PSO, CH/HS, and GA/HS.

For the second problem as shown in Fig. 9, it is one of the sections for a major hydropower project in China where there are 17 layers of soil beneath the dam. The material parameters are shown in Table 2. It is noticed that soil layers 7, 8, 11, 12, 13, and 16 have very low soil parameters, which means that they are the strong local minima which can create problem during optimization search. The nature of this problem is actually similar to that as shown in Fig. 2. The stability of the foundation of the dam is extremely important, as this project is very expensive and the failure of the dam will cause significant loss of lives and properties. Again, many classical optimization methods fail to work well for this case, as there are several layers of soft materials which are strong local minima affecting the direction of search for the global minimum.

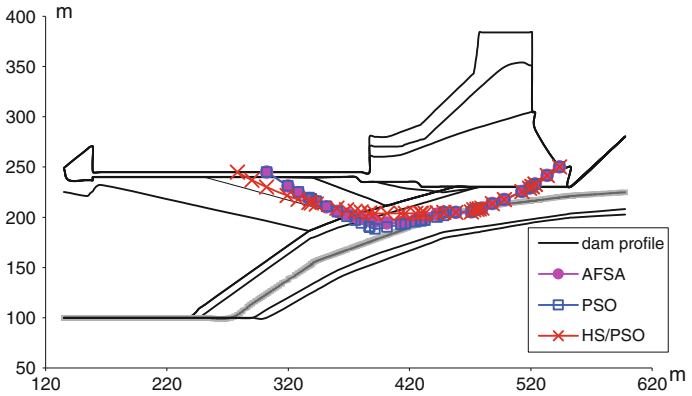

Fig. 9 Critical slip surfaces by different global optimization methods based on the Spencer's method (critical failure surface by MHS and MPSO are not shown for clarity)

Table 2 Geotechnical parameters for the problem in Fig. 9

Layers	γ (kN/m^3)	c′ (kPa)	ϕ′ (°)
1	16.00	2000.0	56.31
2	24.00	2000.0	56.31
3	24.00	2000.0	56.31
4	26.00	1000.0	50.20
5	26.00	1400.0	54.50
6	26.00	1000.0	44.70
7	26.00	100.0	19.30
8	26.00	100.0	19.30
9	26.00	1000.0	44.70
10	26.00	1400.0	54.50
11	26.00	100.0	19.30
12	26.00	100.0	19.30
13	26.00	100.0	19.30
14	23.00	130.0	22.30
15	26.00	1400.0	54.50
16	26.00	100.0	19.30
17	26.00	1400.0	54.50

For the present analysis, the left exit end of the failure surface is searched within the domain of x = 260 to 330 m while the right exit end is searched within the domain of x = 520 to 575 m according to the suggested approach by Cheng [15] and Cheng et al. [7]. It is noticed that the failure surfaces based on the MHS and MPSO are close to the original MHS and MPSO methods and they are not shown for clarity. It is also noticed that the failure surfaces from all the optimization methods are virtually the same at the right-hand side as this is strongly governed by the soil

profiles and the geometry of this project. The major differences between the failure surfaces from different methods of optimization as shown in Fig. 9 are: (1) the starting point of the critical failure surface from HM/PSO is x = 278.0 while it ranges from 320.25 to 320.38 for all the other methods; (2) the exit angle of the failure surface for HM/PSO method is smaller than all the other methods; (3) all the optimization methods except for the HM/PSO is more attracted by soil 13 in the analysis so that the critical failure surfaces are deeper than that by the HM/PSO. In Table 3, it is clear that most of the global optimization methods are not satisfactory except for the AFSA which gives a factor of safety less than 2.0 (but still not good enough while HM, PSO are actually poor in performance) but requires 394,527 trials in the analysis. Actually, when the number of control variable is large, the authors [8] found that HS can be inefficient and sometimes noneffective. It can be viewed that all the optimization methods are attracted by the presence of the "strong" local minima during the search, except for the hybrid HM/PSO, CH/HS, and GA/HS analyzes which are less affected by the "attraction" of the local minima. Based on the proposed coupling method, the minimum factor of safety is obtained as 1.65 with the three hybrid optimization methods. It is noticed that the hybrid optimization methods are more stable for problems where there are several "strong" local minima. For the present large-scale construction work, a good result is much more important than the time of computation, and the proposed hybrid method has provided a good result without excessive computations.

The author has considered another case which is illustrated in Fig. 10. The cohesive strengths of the three soils in Fig. 10 are 28.5 kPa, 0, and 28.5 kPa, respectively, while the friction angles are 20°, 10°, and 20°, respectively (as measured from top to bottom). In this problem, the thickness of soil layer 2 is only 0.1 m as compared with the width of the potential slip surface which is about 30 m. This problem is more difficult than the one in Fig. 9 because the thickness of the soft layer is extremely smaller, and this layer will control the failure. Unless the domain transformation technique is applied, most of the heuristic optimization algorithms will fail to obtain the critical failure surface (shown in Fig. 10) and the minimum factor of safety (0.7603) automatically. For the present problem, the results of analysis are shown in Table 4. It is noticed that only the hybrid method can provide good solutions for the present problem. The special feature of this problem is that a very minor change in the location of the trial failure surface can have a great influence in the factor of safety, and there are many discontinuous solutions during the optimization process. Unless the thickness of soil layer 2

Table 3 Minimum factors of safety for Fig. 9 based on Spencer method (41 control variables)

Method of global optimization	PSO	MPSO	AFSA	MHM	HM/PSO	CH/HS	GA/HS
Min. FOS	2.18	2.15	1.83	1.98	1.65	1.65	1.65
No. of trials	121,124	59,288	394,527	132,098	130,156	123,063	133,238
Min. FOS at evaluation number	99,824	35,460	219,284	98,426	112,342	112,436	120,347

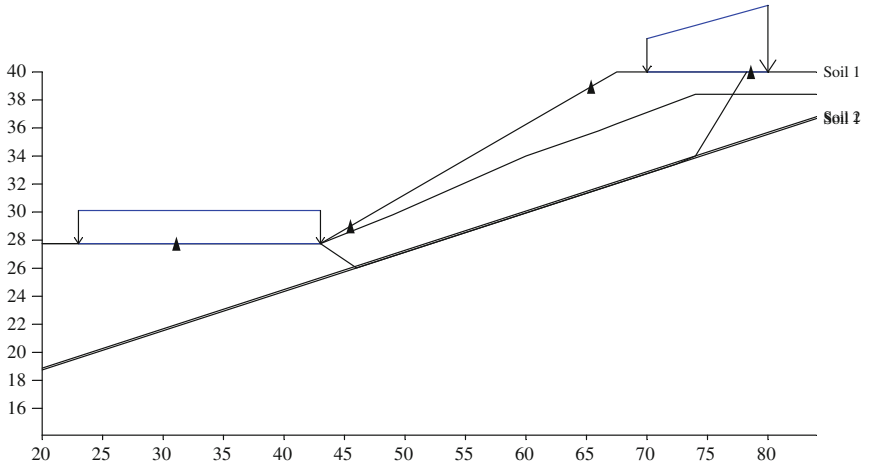

Fig. 10 A difficult problem illustrating the performance of various optimization algorithms, with the critical failure surface as shown

Table 4 Minimum factor of safety for different methods without using the domain transformation technique by Cheng [5]

Method	Min. FS
SA	0.8993
HS	0.9453
MHS	0.805
PSO	0.864
MPSO	0.903
GA	0.979
Ant colony	1.068
HS/PSO	0.770
CH/HS	0.765
GA/HS	0.766

increases to 0.4 m, the classical heuristic optimization algorithms fail to perform satisfactorily. That means, even if an engineer adopts different basic heuristic optimization algorithms, he may still fail to obtain the critical solution which is 0.7603. It is true that problems similar to that in Fig. 10 are not common, but there are some similar cases in Hong Kong and other countries.

For the present problem, the author has found that there are many local minimum within the solution domain, and the factor of safety is very sensitive to the precise location of the trial failure surface. A minor change in the location of the trial failure surface can give rise to noticeable change in the objective function, and the present problem is a typical difficult global optimization problem. For example, if ant colony method is used, a factor of safety 0.903 is obtained with the optimum solution as shown in Fig. 11. If the harmony search method is used, the factor of safety will be 0.899. The critical failure surfaces from these two methods also differ

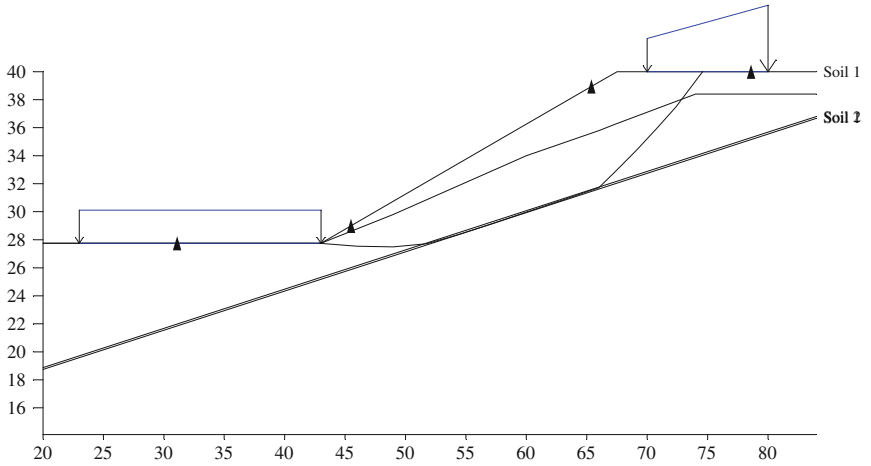

Fig. 11 A local minimum is obtained by the ant colony method for the problem in Fig. 10

noticeably from the critical solution by the hybrid optimization method as shown in Fig. 10. It is of course possible to tune the parameters of the various simple optimization methods to give the results in Fig. 10, but such tuning is not possible unless the critical solution is known. If the thickness of the soft band is reduced to 0.1 mm, which is equivalent to a very sharp change in the soil parameters within a very narrow region, then no optimization method can give the results in Fig. 10 unless the domain transformation technique or equivalently a random function with more weighting for the soft band zone is used. This case can also illustrate the

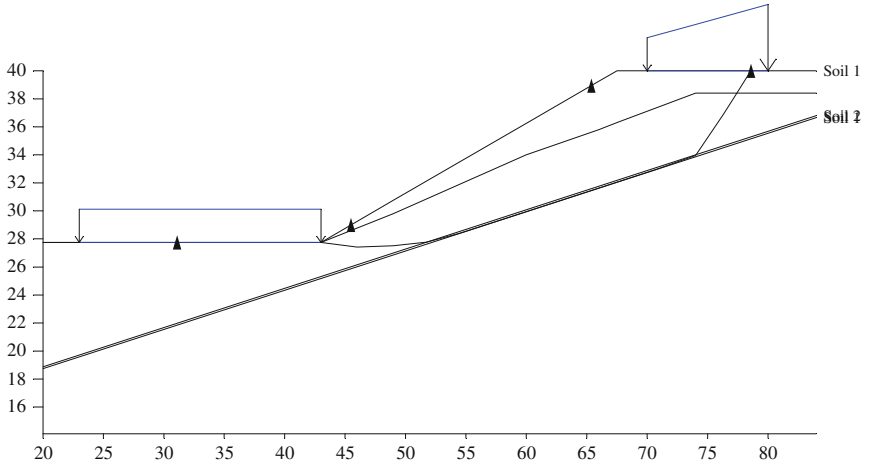

Fig. 12 Another local minimum obtained by the harmony search method for the problem in Fig. 10

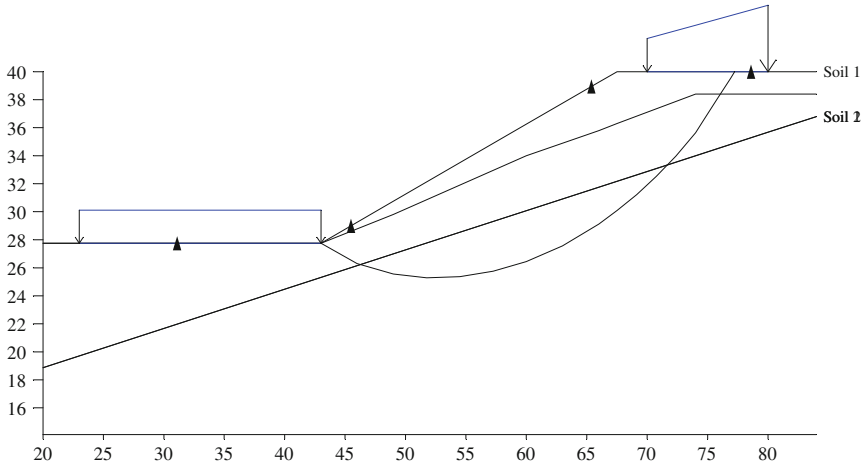

Fig. 13 All optimization methods fail to obtain the critical solution for a soft band thickness of 0.1 mm

limitations of all the heuristic optimization methods that they may not be able to function well if there is an extremely sharp change within a very narrow region. The limitations of the heuristic or other type of optimization methods should always be aware of during application, and it is well known that there is no universal method which can solve all optimization problems under all conditions (Figs. 12 and 13).

9 Discussion and Conclusion

Many geotechnical and other engineering problem can be formulated as an optimization problem, for example, the stability-type problems. There are also many engineering problems which are governed by partial differential equations but can also be formulated as the extremum of a functional. Although many classical problems can be solved by the use of simplex or gradient methods, there are also many types of problems for which many local minima exist in the solution domain. Currently, many researchers are now turning to the use of heuristic optimization methods, which are not limited by the presence of a local minimum during the optimization process.

The author has carried out many studies toward the use of various heuristic optimization methods in geotechnical problems, and it is established that no single heuristic global optimization problem can outperform other methods under all cases. Furthermore, it is difficult to design a set of parameters which are suitable for all types of conditions. Every method has its own merits and limitations, and even automatic tuning of the parameters may only be suitable for some types of problems. For very complicated problems, particularly for those discontinuous problem

or problems with several strong local minima, many simple heuristic optimization methods may sometimes fail to give the best solution. In views of that, the author has tried to combine the use of two heuristic optimization methods to give a more stable and better performance method with less dependence on the choice of parameters. The hybrid methods of PSO/HS, CH/HS, and GA/HS as proposed in this chapter have taken the advantages of two optimization methods so as to give a more stable global optimization. As demonstrated in the present chapter, the hybrid method can perform better than the simple heuristic optimization methods for complicated problems where there are many strong local minima and discontinuous objective functions. From many practical applications, the author has also noticed that the increase in the computations as compared with the simple heuristic optimization methods is not major. It is true that the hybrid method is less efficient (still effective) for many simple problems and is not recommended for such cases. For those difficult problems where there are many local minima, the use of the hybrid methods are however more effective at the expense of minor increase in the amount of computations. The author views that effectiveness and efficiency cannot be maintained simultaneously for arbitrary problems. The proposed hybrid methods are simple enough to be adopted while the performance is more stable than the simple methods for the difficult problems. Finally, the author would also like to address that every method may fail to work under some very special case, and such special case is usually obvious so that the engineers can adopt some special trick (for example, the domain transformation method by Cheng [5]) to overcome the very special constraints. Engineers should not rely completely on the optimization methods without an evaluation about the nature of the problem.

Acknowledgments The author would like to thanks the Hong Kong Polytechnic University for the support of present work through project account YBBY.

References

1. Zienkiewicz O.C., Taylor R.L., Zhu J.Z.: The Finite Element Method: its Basis and Fundamental, 6th edn. Elsevier, New Work (2005)
2. Denn M.M.: Optimization by Variational Methods. Hills Publishing
3. Cheng, Y.M., Li, L., Fang, S.S.: Improved harmony search methods to replace variational principle in geotechnical problems. J. Mech. **27**(1), 107–119 (2011)
4. Cheng, Y.M., Li, D.Z., Li, N., Li, Y.Y., Au, S.K.: Solution of some engineering partial differential equations governed by the minimal of a functional by global optimization method. J. Mech. **29**(3), 493–506 (2013)
5. Cheng, Y.M.: Global optimization analysis of slope stability by simulated annealing with dynamic bounds and Dirac function. Eng. Optim. **39**(1), 17–32 (2007)
6. Cheng, Y.M., Li, L.: Particle swarm optimization algorithm for non-circular failure surface in two dimensional slope stability analysis. Comput Geotech. **34**(2), 92–103 (2007)
7. Cheng, Y.M., Li, L., Chi, S.C.: Studies on six heuristic global optimization methods in the location of critical slip surface for soil slopes. Comput. Geotech. **34**, 462–484 (2007)

8. Cheng, Y.M., Li, L., Chi, S.C., Wei, W.B.: Determination of critical slip surface using artificial fish swarms algorithm. J. Geotech. Geoenvironmental Eng. ASCE **134**(2), 244–251 (2008)
9. Cheng, Y.M., Li, L., Lansivaara, T., Chi, S.C., Sun, Y.J.: Minimization of factor of safety using different slip surface generation methods and an improved harmony search minimization algorithm. Eng. Optim. **40**(2), 95–115 (2008)
10. Fleming, K., Weltman, A., Randolph, M., Elson, K.: Piling Engineering, 3rd edn. Taylors and Francis (2009)
11. Rausche, F., Likins, G. E., Liang, L., Hussein, M.H.: Static and Dynamic Models for CAPWAP Signal Matching. The Art of Foundation Engineering Practice. In: Hussein, M.H., Anderson, J.B., Camp W.M. (eds.), American Society of Civil Engineers, Reston, pp. 534–553. Geotechnical Special Publication No. 198, VA (2010)
12. Greco, V.R.: Efficient monte carlo technique for locating critical slip surface. J. Geotech. Eng. ASCE **122**, 517–525 (1996)
13. Malkawi, A.I.H., Hassan, W.F., Sarma, S.K.: Global search method for locating general slip surface using Monte Carlo techniques. J. Geotech. Geoenvironmental Eng. **127**, 688–698 (2001)
14. Zolfaghari, A.R., Heath, A.C., McCombie, P.F.: Simple genetic algorithm search for critical non-circular failure surface in slope stability analysis. Comput. Geotech. **32**, 139–152 (2005)
15. Cheng, Y.M.: Locations of critical failure surface and some further studies on slope stability analysis. Comput. Geotech. **30**, 255–267 (2003)
16. Cheng, Y.M., Li, N. (2010), A coupled global optimization algorithm for difficult problems, The young Southeast Asian Geotechnical Conference, Taipei, Taiwan, May 9–12, p. 22
17. Cheng, Y.M., Li, L., Sun, Y.J., Au, S.K.: A coupled particle swarm and harmony search optimization algorithm for difficult geotechnical problems. Struct. Multi. Optim. **45**, 489–501 (2012)
18. Cheng, Y.M., Lansivaara, T., Baker, R., Li, N.: The use of internal and external variables and extremum principle in limit equilibrium formulations with application to bearing capacity and slope stability problems. Soils Found. **53**(1), 130–143 (2013)
19. Wang, L., Liu, B.: Particle Swarm Optimization and Scheduling Algorithms. Tsinghua University Press, China (2008)
20. Cheng, Y.M., Lau, C.K.: Soil Slope Stability Analysis and Stabilization—New Methods and Insights, 2nd edn. Spon Press, (2014)
21. Kahatadeniya, K.S., Nanakorn, P., Neaupane, K.M.: Determination of the critical failure surface for slope stability analysis using ant colony optimization. Eng. Geol. **108**, 133–141 (2009)