The Islamic University of Gaza
High Studies Deanery
Faculty of Engineering
Civil Engineering Department
Design and Rehabilitation of Structures

الجامعة الإسلامية بغزة
عمادة الدراسات العليا
كلية الهندسة
قسم الهندسة المدنية
برنامج تصميم وتأهيل المنشآت

Design Optimization of Reinforced Concrete Frames using Artificial Bee Colony Algorithm

By

Mahmoud Maher Jahjouh

Supervised By

Dr. Mohammed Arafa

Dr. Mamoun Al-Qedra

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Civil Engineering – Design and Rehabilitation of Structures

1433هـ - 2012م

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

عمادة الدراسات العليا

هاتف داخلي: 1150

الرقم ج س غ/35/ ....Ref

التاريخ 2012/04/29..........Date

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة الدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ **محمود ماهر محمود ججحوج** لنيل درجة الماجستير في **كلية *الهندسة*/قسم الهندسة المدنية-تصميم وتأهيل منشآت** وموضوعها:

Design Optimization of Reinforced Concrete Frames using Artificial Bee Colony Algorithm

وبعد المناقشة العلنية التي تمت اليوم الأحد 08 جماد آخر 1433هـ، الموافق 2012/04/29م الساعة العاشرة صباحاً بمبنى القدس، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. محمد حسني عرفة	**مشرفاً ورئيساً**	....
د. مأمون عبدالحميد القدرة	**مشرفاً**	....
أ.د. سمير محمد شحادة	**مناقشاً داخلياً**	....
د. نصر الدين صادق المزيني	**مناقشاً خارجيًا**	....

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في **كلية *الهندسة*/ قسم الهندسة المدنية-تصميم وتأهيل منشآت.**

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق ،،،

عميد الدراسات العليا

أ.د. فؤاد علي العاجز

ABSTRACT

The objective of this study is to develop an optimization model that is capable of obtaining the optimum design for reinforced concrete frames in terms of cross section dimensions and reinforcement details. The optimization is carried out using Artificial Bee Colony (ABC) Algorithm, while still satisfying the strength and serviceability constraints of the American Concrete Institute Building Code Requirements for Structural Concrete and Commentary (ACI318M-08).

The Artificial Bee Colony (ABC) Algorithm was recently developed by Karaboga(2005) based on the foraging behavior of a honey bee swarm. The ABC algorithm has proved itself as a reliable and robust optimization algorithm in various optimization problems ranging from numerical functions to the optimization of steel trusses.

A broader reinforcement detailing scheme was utilized in this study when compared with the previous studies conducted on similar topics: Cut off bars were utilized in beams to reduce overall cost. Additional considerations were taken into account such as joint detailing, shear design as well as various column reinforcement arrangements.

Three case studies were considered. The first case was a frame of one bay and one story and had a design space of 7.46×10^{13} possible frame designs and was used as a test frame to obtain the best combination of the ABC algorithm control parameters. Consequently, two frames were studied: a three bay four story frame with a design space of 4.13×10^{36} possible frame designs, and a three bay eight story frame with a design space of 2.98×10^{52} possible frame designs.

The results for the three bay four story frame and three bay eight story were compared with a study previously conducted by other researchers who used two different optimization algorithms, namely The heuristic big bang-big crunch (HBB-BC) algorithm, which is based on big bang-big crunch (BB-BC) and a harmony search (HS) scheme to deal with the variable constraint, and The (HPSACO) algorithm, which is a combination of particle swarm with passive congregation (PSOPC), ant colony optimization (ACO), and harmony search scheme (HS) algorithms. The results prove that the ABC algorithm as well as the design variables used in the ABC yielded better results than the previous study: For the three bay four story frame, cost savings of (5.5%) were achieved whereas for the three bay eight story frame, cost savings of (3.5%) were achieved.

الملخص

تهدف الرسالة لتطوير نموذج تصميمي يستطيع الوصول إلى التصميم الأمثل للإطارات الخرسانية المسلحة من حيث أبعاد المقاطع وتفاصيل التسليح لعناصر الإطار. تم استخدام طريقة مستعمرة النحل الاصطناعية في إيجاد الحل الأمثل والذي يخضع ويلبي جميع متطلبات القوة والمتانة حسب متطلبات الكود الأمريكي للتصميم (ACI318-08).

طورت طريقة مستعمرة النحل الاصطناعية حديثاً عبر الباحث كارابوجا، وتعتمد على محاكاة تصرف جماعات النحل في بحثها عن الغذاء، وقد أثبتت هذه الطريقة نفسها في مجالات عديدة كإيجاد قيم دوال رياضية صعبة، وإيجاد التصميم الأمثل للجمالونات المعدنية.

تفاصيل التسليح المستخدمة في هذه الرسالة أوسع من تلك المستخدمة في الدراسات السابقة المماثلة: كاستخدام حديد الموازين في الجوائز لتخفيض التكلفة الأجمالية، كما أخذت الدراسة الحالية العديد من القضايا التصميمية الإضافية في عين الاعتبار: كتفاصيل تسليح الوصلات وتصميم العناصر لقوى القص والطرق المختلفة المستخدمة في تفصيل حديد الأعمدة.

تم دراسة ثلاثة إطارات في هذه الرسالة، أولها إطار خرساني ذو جائز واحد وطابق واحد، والذي يحتمل 7.46×10^{13} احتمالاً تصميمياً. تم استغلال هذا الإطار الخرساني للوصول لأفضل توليفة من معاملات التحكم الخاصة بطريقة مستعمرة النحل الاصطناعية. قامت هذه الدراسة أيضاً بتناول إطار خرساني ذو ثلاثة جوائز وأربعة طوابق، والذي يحتمل 4.13×10^{36} احتمالاً تصميمياً. واختتمت الدراسة بإطار خرساني ذو ثلاثة جوائز وثمانية طوابق، والذي يحتمل 2.98×10^{52} احتمالاً تصميمياً.

نتائج الإطار الثاني والثالث تم مقارنتها بدراسة سابقة أجريت مؤخراً، واستُخدِمَ فيها طريقتان مختلفتان للوصول إلى الحل الأمثل في حل الإطار الثاني والثالث. من خلال المقارنة تبين أن طريقة مستعمرة النحل الاصطناعية والمتغيرات التصميمية المستخدمة في هذه الدراسة أثبتت أنها أكثر كفاءةً من تلك المستخدمة في الدراسة السابقة، حيث استطاعت الدراسة الحالية أن تحقق تقليصاً في تكلفة الإطار الثاني بمقدار (5.5%)، كما استطاعت أن تحقق تقليصاً مقداره (3.5%) للإطار الثالث.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisors, Dr. Mohammed Arafa and Dr. Mamoun Al-Qedra for their guidance and enormous efforts during the project. I would like to express my deepest gratitude to both of them for their generosity and support from the beginning to end.

I would like to give my special thanks to my beloved father: Eng. Maher Jahjouh as well as my mother, brothers, sisters and friends who have supported me all the way.

I also extend my heartiest gratitude to my wife, for her constant encouragement and advice. This thesis would not have been possible without her patience and support.

Last but not least, I wish also to acknowledge Prof. Mehmet Polat Saka from the University of Bahrain for his constant encouragement and advice as well as Prof. Ali Kaveh from the Iran University of Science and Technology for his generous guidance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

ABC	: Artificial Bee Colony
RC	: Reinforced Concrete
GA	: Genetic Algorithm
HS	: Harmony Search
ACI	: American Concrete Institute
SA	: Simulating Annealing
ACO	: Ant Colony Optimization
PSO	: Particle Swarm Optimization
HBB-BC	: Heuristic Big Bang-Big Crunch
BB-BC	: Big Bang-Big Crunch
HPSACO	: Combination of Particle Swarm with Passive Congregation, Ant Colony Optimization and Harmony Search
PSOPC	: Particle Swarm with Passive Congregation
N_S	: Number of food sources to be investigated by bees
D	: Dimension of each solution vector (the number of variables in a solution vector)
SDC	: Seismic Design Category
$\emptyset$	: Strength reduction factor
R_n	: Nominal resistance of reinforced concrete member
R_u	: Ultimate applied external load
W	: Wind Loads
E	: Seismic Loads
D (load)	: Dead Load
L	: Live Load
E_c	: Modulus of Elasticity (MPa)
f_c'	: Specified compressive strength of concrete (MPa)
I	: Moment of inertia (mm^2)
I_g	: Moment of inertia of gross cross section (mm^4)
UBC	: Uniform Building Code
$\Delta_{rel.max}$	: Maximum permissible relative drift between two adjacent stories (m)
H	: Story Height (m)
Q	: Stability Index

$\sum P_u$	: Total factored vertical load for all of the columns on the story in question (kN)
Δ_0	: The elastically determined first order lateral deflection due to shear with respect to the bottom of that story (mm)
V_u	: the total factored horizontal shear for the story in question (kN)
l_c	: The height of a compression member in the frame measured from center to center of frame joints (mm)
c	: Depth of neutral axis (mm)
a	: Depth of equivalent rectangular compressive block of concrete (mm)
β_1	: Relation between depth of neutral axis to depth of rectangular stress block
C (force)	: Compressive force in beam section (N)
T (force)	: Tensile force in beam section (N)
A_s	: Total area of flexural steel in tension zone (mm^2)
f_y	:Yielding strength of reinforcement (MPa)
b	: Width of cross section (mm)
d	: distance from the center of tension reinforcement to the extreme compression fiber (mm)
M_n	: Nominal flexure capacity of beam (kN.m)
l_d	: Development length (mm)
ψ_t	: Factor used to modify development length based on reinforcement location
ψ_s	: Factor used to modify development length based on reinforcement size
ψ_e	: Factor used to modify development length based on reinforcement coating
c_b	: Smaller of distance from center of bar to concrete surface and half of center to center spacing between bars (mm)
K_{tr}	: Transverse reinforcement index
d_b	: Diameter of reinforcement bar being developed (mm)
A_{tr}	: Area of transverse reinforcement (mm^2)
s	: maximum center-to-center spacing of transverse reinforcement within development length (mm)
n (eq 4.15)	: number of bars being developed along plane of splitting
V_n	: Nominal shear strength of a reinforced concrete beam section (kN)
V_c	: Nominal shear strength provided by concrete (kN)
V_s	: Nominal strength provided by shear reinforcement (kN)
λ	: Modification factor reflecting the reduced mechanical properties of lightweight concrete

b_w	: width of beam web (mm)
A_v	: Area of transverse reinforcement (mm^2)
f_{yt}	: yield strength of transverse reinforcement (MPa)
P_n	: Nominal axial force resistance (kN)
ϵ_{cu}	: Crushing strain of concrete
ϵ_s	: Tensile strain of steel layer
d_t	: Depth of extreme steel layer in columns (mm)
ϵ_t	: Strain of extreme steel layer in columns
ϵ_y	: Yield strain of steel reinforcement
Z	: Constant that controls the multiple points on the interaction diagram
E_s	: The elastic modulus of reinforcing steel bars (MPa)
C_c	: Compressive force of concrete in reinforced concrete column (N)
F_{si}	: Force in steel layer in reinforced concrete column (N)
$\emptyset'$	: Reduction factor that accounts for accidental eccentricities
A_g	: Gross area of cross section (mm^2)
k	: Effective length factor
l_u	: Unbraced length of columns (m)
r	: Radius of Gyration (m)
M_1	: Smaller end moment on column (kN.m)
M_2	: Larger end momen on column (kN.m)
ψ	: Relative stiffness between columns and beams
M_c	: Magnified design moment (kN.m)
δ_{ns}	: Non-sway moment magnification coefficient
C_m	: Factor relating actual moment diagram to an equivalent uniform moment diagram
P_c	: Critical buckling load (kN)
β_{dns}	: Ratio of maximum factored axial sustained load to maximum factored axial load associated with the same load combination
$M_{2,min}$	: Minimum applied moment on long columns (kN)
M_{2ns}	: Factored bending moment resulting from non-sway loads (kN.m)
δ_s	: Sway moment magnification factor
M_{2s}	: Factored bending moment resulting from sway loads (kN.m)
l_{dh}	: Anchorage of bars developed in joints (mm)

$\Delta_{rel.max}$	: Maximum permissible relative drift between two adjacent Stories (m)
H	: Story Height (m)
$A_{st,min}$	: Minimum reinforcement area in columns (mm^2)
$A_{st,max}$	: Maximum reinforcement area in columns (mm^2)
b_{top}	: Width of top column (mm)
b_{bottom}	: Width of bottom column (mm)
h_{top}	: Length of top column (mm)
h_{bottom}	: Length of bottom column (mm)
b_{beam}	: Width of beam (mm)
b_{column}	: Width of column (mm)
h_{column}	: Length of column (mm)
$V_{s,max}$	: Maximum shear force carried by shear reinforcement (kN)
S_{min}	: Minimum clear spacing between flexural bars in beams (mm)
$Max.Agg.S$	: Maximum Size of Aggregate in concrete mix (mm)
N_{Span}	: Number of spans in frames
N_{cb}	: Number of continuous bottom reinforcement bars
d_{cb}	: Diameter of continuous bottom reinforcement bars (mm)
N_{pb}	: Number of cutoff bottom reinforcement bars
d_{pb}	: Diameter of cutoff bottom reinforcement bars (mm)
N_{ct}	: Number of continuous top reinforcement bars
d_{ct}	: Diameter of continuous top reinforcement bars (mm)
N_{pt}	: Number of cutoff top reinforcement bars
d_{pt}	: Diameter of cutoff top reinforcement bars (mm)
d_s	: Diameter of stirrup bars (mm)
b_c	: Width of column (mm)
h_c	: Length of column (mm)
n_b	: Number of bars in colun
d_b	: Diameter of column bars (mm)
N_{CG}	: Number of Column Groups
N_{BG}	: Number of Beam Groups
$F(x)$	: Unconstrained objective function
F_b	: Total cost of beams ($)

F_c	: Total cost of columns ($)
N_b	: total number of beams in the frame
N_c	: total number of columns in a frame
C_c (cost)	: Cost of concrete ($/m^3)
C_s	: Cost of steel ($/kN)
γ_s	: Unit weight of steel (kN/m^3)
V_{it}	: total volume of member (m^3)
V_{is}	: volume of steel reinforcement in a member (m^3)
A_f	: total formwork area (m^2)
C_f	: Cost of formwork ($/m^2)
C (constraint)	: Constraint violation function
$\sum_{i=1}^{n} c_i$	: The total of all constraint violations
$\varphi(x)$	: Penalized objective function
K	: Penalty function constant
ϵ	: Penalty function
N_P	: Number of Bees in the Colony
I_l	: The improvement limit for a solution
I_{max}	: Maximum number of iterations
VCP	: Variable Changing Percentage to derive a mutant solution
r	: Number of independent runs

CHAPTER 1: INTRODUCTION

1.1 General

Reinforced concrete structures have considerable compressive strength compared to most other materials. In addition to the high compressive strength, reinforced concrete structures are durable, versatile, and have relatively low maintenance cost when compared to steel structures. They also provide good resistance against fire and water damage, and have excellent potential for long service life (Wight & MacGregor, 2008).

Material cost is an important issue in the design and construction of reinforced concrete structures. The main factors affecting cost are the amount of concrete and steel reinforcement required. It is, therefore, desirable to make reinforced concrete structures lighter, while still fulfilling serviceability and strength requirements. In addition to material cost, labor and formwork costs are significant. Good engineers are those capable of designing low cost structures without compromising its function or violating structural constraints.

The traditional approach to design reinforced concrete members does not fully optimize the use of materials. Most designs are based on the prior experience of the engineer, who selects cross-section dimensions and material grades by comparing past experience. This gives rise to fixed rules-of-thumb for preliminary designs (Zaforteza et. al., 2009). This process is typically of high cost in terms of time, human effort and material usage, which makes structural optimization procedures using artificial intelligence a clear alternative to designs based on experience (Coello, 1997).

Optimization of reinforced concrete members is a complex problem, due to the large number of variables that influence the design process, the different nature of these variables and the various reinforcement details available for a single design problem.

The optimization technique used in the research work of this thesis is the Artificial Bee Colony (ABC) Algorithm, which is an optimization algorithm based on the intelligent behavior of a honey bee swarm (Karboga, 2005). In ABC algorithm, the colony of artificial bees consists of three groups of bees: employed bees, onlookers and scouts who search for better food sources that correspond to better frame designs.

1.2 Problem Statement

Obtaining an optimal solution within a large space of possible solutions is very complex to solve by hand, and even traditional approaches fail in obtaining such solution. This is due to the large number of design variables, their interaction with each other and their influence on the final cost. Typically, the design is limited by some constraints such as the choice of material, required strength, displacements, loads, support conditions and achieving requirements as stated in codes of practice.

The optimization of reinforced concrete (RC) members is very complicated due to the absence of standard RC sections like those in its steel counterpart. Furthermore, RC sections deal with both discrete and continuous variables. Moreover, a large number of possible reinforcement detailing can still achieve the strength and serviceability required. The large number of detailing possibilities adds more complications to the problem at hand.

This research considers an approach based on the Artificial Bee Colony (ABC) Algorithm for optimizing the cross section and reinforcement details of reinforced concrete frames, with discrete design variables. The ABC algorithm has proved to be a robust and efficient optimization technique capable of competing with some of the most famous optimization techniques such as the Genetic Algorithm (GA) and Harmony Search (HS) Algorithm (Hadidi & Kazemzadeh, 2010).

1.3 Motivation

Design optimization methods have been used to obtain more economical designs since 1970s (Pincus, 1970) - (Glover, 1977). Numerous algorithms have been developed for accomplishing the optimization problems in the last four decades. The early works on the topic mostly use mathematical programming techniques or optimality criteria with continuous design variables. These methods utilize gradient of functions to search the design space.

Today's competitive world has forced the engineers to realize more economical designs and designers to develop more effective optimization techniques. As a result, heuristic search methods emerged in the first half of 1990s (Jenkins, 1991)

Karaboga (2005) proposed the ABC algorithm inspired by the foraging behavior of honeybees, which has proved to be a robust optimization algorithm in the process of obtaining optimal and near optimal solutions for steel trusses (Hadidi & Kazemzadeh, 2010).

The constant search for optimality and the new promising optimization technique (ABC), the lack of research in the field of optimization of reinforced concrete structures, as well as the challenge of being one of the first researchers to work in such hard field were the main driving force that led to the research works presented in this thesis.

1.4 The Objective of This Research

The main objective of the current study is to develop an optimization model that is capable of obtaining the optimum design for reinforced concrete frames in terms of cross section dimensions and reinforcement details. The optimization is carried out using Artificial Bee Colony (ABC) Algorithm, while still satisfying the strength and serviceability constraints of the American Concrete Institute Building Code Requirements for Structural Concrete and Commentary (ACI318M-08). This model is then applied to study cases to obtain results and draw possible conclusions and recommendations. The objectives of this study are:

1) Develop a computer model which designs reinforced concrete frames according to the ACI strength and serviceability constraints.
2) Establish an Artificial Bee Colony Algorithm that interacts with the developed computer design model
3) Carry out validation and verification of the developed models.
4) Compare the optimization results with previous studies
5) Draw conclusions and recommendations

1.5 Research Scope and Limitations

The scope of the current study includes:

1) Linear behavior of RC frames
2) Two dimensional planar frames
3) Design conforms to the strength and serviceability constraints of the (ACI318M-08)
4) Optimization process includes optimizing both cross sectional dimensions and steel details for beams, columns and joints in the frame. Foundations are considered out of scope of this study.
5) Optimization of reinforcement is limited to main reinforcement, namely: shear, flexure and axial reinforcement. Secondary reinforcement such as confining U-bars and skin reinforcement are considered out of scope.

1.6 Methodology

To achieve the objectives of this research, the following tasks were executed:

1) Conducting a literature review about optimization techniques, optimization of RC frames and constraints applied in previous studies.
2) Developing of model using Matlab to analyze and design reinforced concrete frames conforming to strength and serviceability constraints of the ACI code.
3) Establishing a suitable Artificial Bee Colony Algorithm that interacts with the design model.
4) Validating the developed computer design optimization model using several frames.
5) Performing optimization and verifying results.
6) Comparing results with previous studies.
7) Drawing conclusions and recommendations.

1.7 Contents of the Thesis

This thesis comprises 7 chapters and 3 appendices, which are formulated in a logical sequence for the reader to follow. Chapter 2 discusses the main concepts of optimization as well as some of the latest optimization techniques used and is then ended by listing the latest research done in the field of reinforced concrete optimization. Chapter 3 is devoted for the discussion of the Artificial Bee Colony (ABC) technique. It discusses the basics of swarm intelligence, the background of the ABC algorithm as well as the main control parameters of the algorithm. Chapter 4 discusses the design and analysis of reinforced concrete frames conforming to the ACI318M-08 code. Chapter 5 discusses the formulation of optimization problem, the objective function, penalty functions and optimization program flowchart. Chapter 6 analyzes three reinforced concrete frames and presents the results and its analysis. Chapter 7 draws the recommendations and conclusions of the research work of this thesis. Appendix A contains the main code used in Matlab for the optimization works done in this thesis. Appendix B presents a user manual for the program developed for the optimization of reinforced concrete frames using the ABC algorithm. Finally Appendix C contains full and detailed results of all runs that were performed on the one bay one story reinforced concrete frame discussed later on.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction to Optimization

The sections that follow discuss the main concepts of optimization, some characteristics of an optimization problem and various optimization techniques in order to familiarize the reader with the basics of optimization.

2.1.1 Concept of Optimization

Optimization is the process of obtaining the best alternative from a set of possible available alternatives (Stutzle, 2010). Hence it is a system, shown in Figure (2.1), that relies on available alternatives and constraints as input, processes these inputs utilizing an optimization technique and results in the optimum solution as an output.

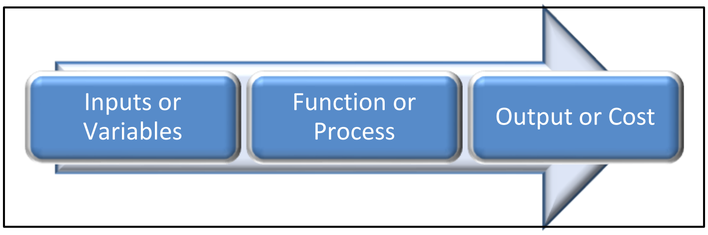


Figure 2. 1: Optimization process as a system

2.1.2 Difference between Optimization and Root Finding

Approaches to optimization are somehow akin to root or zero finding methods, but harder. Bracketing the root or optimum is a major step in hunting it down (Haupt, 2004). For the single-variable case, finding one positive point and one negative point brackets the zero. On the other hand, bracketing a minimum requires three points, with the middle point having a lower value than either end point. In the mathematical approach, root finding searches for zeros of a function, while optimization finds zeros of the function derivatives. Finding the function derivative adds one more step to the optimization process. Many times the derivative does not exist or is very difficult to find, this is the case in many engineering problems where a handful of variables exist and were the relation between variables cannot be mathematically expressed, thus other techniques are required to obtain the optimal solution without mathematical complexities. Those methods will be discussed in section (2.2).

2.1.3 Global Optimality versus Local Optimality

Another difficulty in optimization is determining if a given minimum is the best (global) minimum or a suboptimal (local) minimum (Haupt, 2004). This is of special importance in reinforced concrete frame optimization problems since the number of possible variable combinations for the simplest of reinforced concrete frame are practically infinite. These complications yield several local optimal solutions with one of them being the best, i.e. the global optimum.

A well balanced optimization algorithm should have both local and global search methods. In other words, it should search the vicinity of solutions as well as taking into consideration that the exploration of different places in the search space is equally important.

Figure (2.2) further illustrates this concept. For the simplest mathematical optimization problem of continuous functions, one can find more than one local minima (the white stars) in the domain of the function. One of these is considered the global optimum (the black star), which the algorithm seeks to find (Valle et. al., 2009).

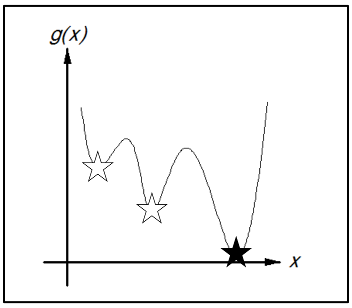


Figure 2. 2: Difference between global optimum and local optimum

2.1.4 Categories of Optimization

Optimization techniques can be divided into six categories, shown in Figure (2.3). None of these six categories or their branches is necessarily mutually exclusive. For instance, a dynamic optimization problem could be either constrained or unconstrained. In addition, some of the variables may be discrete and others continuous.

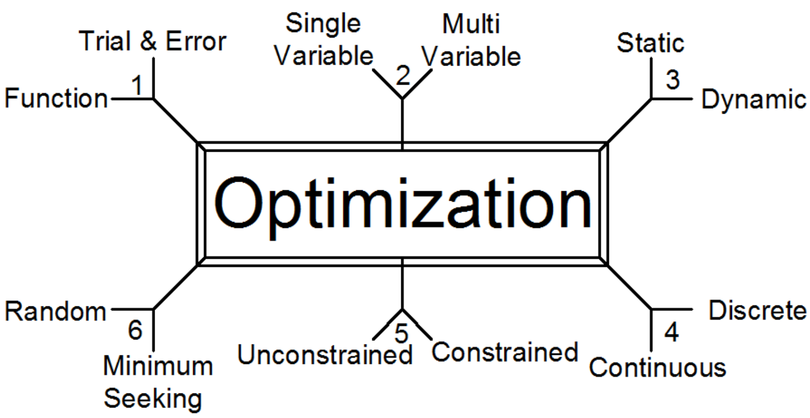


Figure 2. 3: Categories of optimization techniques

The different categories of optimization algorithms are discussed below (Haupt, 2004):

Trial-and-error optimization refers to the process of adjusting variables that affect the output without knowing much about the process that produces the output. In contrast, a mathematical formula describes the objective function in function optimization. Various mathematical manipulations of the function lead to the optimal solution.

If there is only one variable, the optimization is one-dimensional. A problem having more than one variable requires multidimensional optimization. Optimization becomes increasingly difficult as the number of dimensions increases. Many multidimensional optimization approaches generalize to a series of one-dimensional approaches.

Dynamic optimization means that the output is a function of time, while static means that the output is independent of time. For example: Finding the fastest route is a dynamic problem whose solution depends on the time of day, the weather, accidents, and so on. The static problem is difficult to solve for the best solution, but the added dimension of time increases the challenge of solving the dynamic problem.

Optimization can also be distinguished by either discrete or continuous variables. Discrete variables have only a finite number of possible values, whereas continuous variables have an infinite number of possible values. If we are deciding in what order to attack a series of tasks on a list, discrete optimization is employed. Discrete variable optimization is also known as combinatorial optimization, because the optimum solution consists of a certain combination of variables from the finite pool of all possible variables. However, if we are trying to find the minimum value of a function on a number line, it is more appropriate to view the problem as continuous.

Variables often have limits or constraints. Constrained optimization incorporates variable equalities and inequalities into the cost function. Unconstrained optimization allows the variables to take any value. A constrained variable often converts into an unconstrained variable through a transformation of variables. Most numerical optimization routines work best with unconstrained variables.

Some algorithms try to minimize the cost by starting from an initial set of variable values. These minimum seekers easily get stuck in local minima but tend to be fast. They are the traditional optimization algorithms and are generally based on calculus methods. Moving from one variable set to another is based on some determinant sequence of steps. On the other hand, random methods use some probabilistic calculations to find variable sets. They tend to be slower but have greater success at finding the global minimum.

2.1.5 Classical Optimization Techniques

The classical methods of optimization are useful in finding the optimum solution of continuous and differentiable functions. These methods are analytical and make use of the techniques of differential calculus in locating the optimum points. Since some of the practical problems involve objective functions that are not continuous and/or differentiable, the classical optimization techniques have limited scope in practical applications.

The features of the classical optimization techniques as described by (Kumar, 2005) are as follows:

1. These are useful in finding the optimum solution or unconstrained maximum or minimum of continuous and differentiable functions.
2. Make use of differential calculus in locating the optimum solution.
3. Have limited scope in practical applications as some of them involve objective functions which are not continuous and/or differentiable.
4. Assume that the function is differentiable twice with respect to the design variables and the derivatives are continuous.
5. These methods lead to a set of nonlinear simultaneous equations that may be difficult to solve.

The common difficulties with most of the classical optimization techniques are:

1. Convergence to an optimal solution depends on the chosen initial solution.
2. Most algorithms tend to get stuck to a suboptimal solution.
3. An algorithm efficient in solving one search and optimization problem may be not efficient in solving a different problem.
4. Algorithms are not efficient in handling problems having discrete variables.

2.1.6 Heuristic Optimization Techniques

A heuristic technique is a one which seeks good (i.e. near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality (Stutzle, 2010). Heuristic methods typically require far less time than exact methods. Heuristics can be constructive (build a solution piece by piece) or improvement based (take a solution and alter it to find a better solution). The features of the heuristic optimization techniques are described as follows (Stutzle, 2010):

1. The problems are solved iteratively
2. They are capable of optimizing systems which have continuous, discrete or integer design variables
3. The solution is not always the global optimum, this depends on the complexity of the problem
4. The problem does not get trapped in local optimums
5. They do not necessarily produce the same solution each time

2.2 Common Heuristic Optimization algorithms

Over the last four decades, a large number of algorithms have been developed to solve various engineering optimization problems. Most of these algorithms are based on numerical linear and nonlinear programming methods that require substantial gradient information and usually seek to improve the solution in the neighborhood of a starting point. These numerical optimization algorithms provide a useful strategy to obtain the global optimum in simple and ideal models. Many real-world engineering optimization problems, however, are very complex in nature and quite difficult to solve using these algorithms. If there is more than one local optimum in the problem, the result may depend on the selection of an initial point, and the obtained optimal solution may not necessarily be the global optimum. Furthermore, the gradient search may become difficult and unstable when the objective function and constraints have multiple or sharp peaks. The computational drawbacks of existing numerical methods have forced researchers to rely on meta-heuristic algorithms based on simulations to solve engineering optimization problems. The common factor in meta-heuristic algorithms is that they combine rules and randomness to imitate natural phenomena. To solve difficult and complicated real-world optimization problems, however, new heuristic and more powerful algorithms based on analogies with natural or artificial phenomena must be explored. The following sections, give a brief overview of some existing meta-heuristic algorithms.

2.2.1 Genetic Algorithm (GA)

Genetic algorithms (GA) are based on the evolution theory of Darwin. They were proposed by (Holland, 1975). The main principle of GAs is the survival of robust ones and the elimination of the others in a population. GAs are able to deal with discrete optimum design problems and do not need derivatives of functions, unlike classical optimization. However, the procedure for the genetic algorithm is time consuming and the optimum solutions may not be global ones, but they are feasible both mathematically and practically.

2.2.2 Simulating Annealing Algorithm (SA)

Simulating annealing (SA) is an accepted local search optimization method. Local search is an emerging paradigm for combinatorial search which has recently been shown to be very effective for a large number of combinatorial problems. It is based on the idea of navigating the search space by iteratively stepping from one solution to one of its neighbors, which are obtained by applying a simple local change to it. The SA algorithm is inspired by the analogy between the annealing of solids and searching the solutions to optimization problems. SA was developed by Metropolis et. al (1953) and proposed by Kirkpatrick et. al. (1983) for optimization problems.

2.2.3 Ant Colony Optimization Algorithm (ACO)

Ant colony optimization (ACO) is an application of ant behavior to the computational algorithms and is able to solve discrete optimum structural problems. It also has additional artificial characteristics such as memory, visibility and discrete time. ACO was originally put forward by Dorigo et. al. (1992) for optimization problems.

2.2.4 Harmony Search Optimization Algorithm (HS)

Geem and Lee developed a harmony search (HS) meta-heuristic algorithm that was conceptualized using the musical process of searching for a perfect state of harmony (Geem & Lee, 2005). The harmony in music is analogous to the optimization solution vector, and the musician's improvisations are analogous to local and global search schemes in optimization techniques.

2.2.5 The Artificial Bee Colony Algorithm (ABC)

Recently, Karaboga developed a new optimization algorithm called the Artificial Bee Colony (ABC) algorithm (Karboga, 2005). The ABC algorithm was firstly introduced for numerical optimization problems based on the foraging behavior of a honey bee swarm. Further improvements of the ABC algorithm have been carried out by Karaboga and Basturk (2007). In this model, the foraging bees are classified into three different types: employed bees, onlookers and scouts. A bee which has found a food source to exploit is called an employed bee. Onlookers are those waiting in the hive to receive the information about the food sources from the employed bees and Scouts are the bees which are randomly searching for new food sources around the hive.

A number of researches studied and applied the Artificial Bee Colony on several study cases ranging from normal equations to structural design problems. Karaboga and Basturk presented the main outlines of the ABC algorithm (Karaboga & Basturk, 2008). Later on, Akay and Karaboga (2009) applied the ABC algorithm on numerical test functions and compared the results with well-known algorithms such as the GA, Particle Swarm Optimization (PSO) and HS.

2.3 Optimization of Reinforced Concrete

Design optimization of reinforced concrete (RC) structures is challenging because of the complexity associated with reinforcement design. Also, in the case of concrete structures, three different cost components due to concrete, steel and formwork are to be considered and any slight variation in the quantity of any one of these items affects the overall cost of the structure to a great extent (Akin & Saka, 2011). Hence, the problem becomes the selection of a combination of appropriate values of design variables and the quantity of reinforcement so that the total cost component is minimal (Kaveh & Sabzi, 2011).

Kaveh and Sabzi (2011) researched the optimum design of reinforced concrete frames using two different methods: The heuristic big bang-big crunch (HBB-BC), which is based on big bang-big crunch (BB-BC) and a harmony search (HS) scheme to deal with the variable constraint, and The (HPSACO) algorithm, which is a combination of particle swarm with passive congregation (PSOPC), ant colony optimization (ACO), and harmony search scheme (HS) algorithms. They studied three frames and obtained optimum designs of columns and beams without considering joint detailing or shear reinforcement. The design variables used were simply the cross sectional dimensions of columns, column reinforcement, beam cross sectional dimensions as well as the number and diameter of steel bars used as top and bottom reinforcement without using cut off bars

Akin and Saka (2010), studied the optimum detailed design of reinforced concrete continuous beams using the harmony search algorithm. The design variables are selected as the width and the depth of beams in each span, the diameter and the number of longitudinal reinforcement bars along the span and supports and the diameter of ties as well as the number and diameter of cut off bars. The values of these variables are required to be selected from a design pool which contains discrete values for these variables. The design constraints are implemented from ACI 318-05. They also studied the optimum design of concrete cantilever retaining walls using the harmony search algorithm in the same year. In the formulation of the optimum design problem the height and thickness of stem, length of toe projection and the thickness of stem at base level, the length and thickness of base, the depth and thickness of key and the distance from toe to the key are treated as design variables. The design constraints were implemented according to the provisions of ACI 318-05.

Zaforteza et. al. (2009) studied the CO_2 optimization of reinforced concrete frames by simulated annealing. They related the optimum design of a reinforced concrete frame to the amount of CO_2 gas emitted in order to minimize pollution. The design variables define the geometry of the cross sections of beams and columns, the type of steel and concrete as well as the reinforcement of the frame. For reinforcement detailing, they took shear reinforcement and cut off bars into considerations whereas joint detailing was not.

The Optimum design of reinforced concrete plane frames based on predetermined section database was studied by Kwak and Kim (2008). The study formulates a database of all possible cross sections and sorts them according to their strength. In the proposed algorithm, design variables in a RC section such as the sectional dimensions and steel quantity are linked by a single design variable (the section identification number) that removes virtually all of the limitations of mathematical programming methods applied to large complex structures. The authors integrated this technique into genetic algorithms in (2009), and studied an integrated genetic algorithm complemented with direct search for optimum design of RC frames.

Camp et. al. (2003), studied the flexural design of reinforced concrete frames using a Genetic Algorithm. The design variables were the cross section dimensions, reinforcement diameter, number of bars per row and number of rows in a cross section.

Rajeev and Krishnamoorthy (1992) studied the design optimization of reinforced concrete frames using a Genetic Algorithm. The design conforms to the Indian Code of Practice. The design variables were the cross section dimensions, reinforcement diameters, number of bars and rows in a cross section.

2.4 Concluding remarks

Based on the extensive literature review of optimization in general, latest advancements in optimization techniques and the current state of knowledge of optimization of reinforced concrete frames, it can be concluded that the optimization of reinforced concrete frames hasn't been extensively researched yet and that all optimization attempts simplified the design problem as much as possible. Furthermore, a new promising optimization technique called the Artificial Bee Colony (ABC) emerged which proved its efficiency and robustness in various optimization problems such as optimum values for mathematical equations (Karaboga & Basturk, 2008) and the optimization of steel trusses (Hadidi & Kazemzadeh, 2010). Thus this study utilizes the ABC algorithm in the optimization of reinforced concrete frames with broader design variables and considerations.

CHAPTER 3: THE ARTIFICIAL BEE COLONY (ABC) ALGORITHM

3.1 Introduction

The artificial bee colony (ABC) algorithm which was proposed by Karaboga is a novel nature inspired algorithm based on the foraging behavior of a honeybee swarm (Karboga, 2005). Different satisfactory applications of the ABC algorithm have been reported in the literature such as the optimization of steel trusses in (Hadidi & Kazemzadeh, 2010).

3.2 Swarm Intelligence

3.2.1 Relation between ABC algorithm and Swarm Intelligence

The ABC algorithm is a meta heuristic algorithm that relies on swarm intelligence rather than evolutionary procedures. Swarm intelligence has become a research interest to many scientists of related fields in recent years. Bonabeau et. al. (1999) has defined the swarm intelligence as "any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies". Bonabeau et al. focused their viewpoint on social insects alone such as termites, bees, wasps as well as other different ant species. However, the term swarm is used in a general manner to refer to any restrained collection of interacting agents or individuals. The classical example of a swarm is bees swarming around their hive; nevertheless the metaphor can easily be extended to other systems with a similar architecture. An ant colony can be thought of as a swarm whose individual agents are ants. Similarly a flock of birds is a swarm of birds. An immune system (De Castro & Von Zuben, 1999) is a swarm of cells and molecules and crowd is a swarm of people (Vesterstorm & Riget, 2002). Particle Swarm Optimization (PSO) Algorithm models the social behavior of bird flocking or fish schooling (Kennedy & Eberhart, 1995).

3.2.2 Fundamentals of Swarm Intelligence

Two fundamental concepts, self-organization and division of labor, are necessary and sufficient properties to obtain swarm intelligent behavior such as distributed problem solving systems that self-organize and adapt to the given environment (Karboga, 2005):

1) Self-organization can be defined as a set of dynamical mechanisms, which result in structures at the global level of a system by means of interactions among its low-level components. These mechanisms establish basic rules for the interactions between the components of the system. The rules ensure that the interactions are executed on the basis of purely local information without any relation to the global pattern. Bonabeau et al. (1999) have characterized three basic properties on which self-organization relies: Positive feedback, negative feedback and fluctuations:

a. Positive feedback is a simple behavioral “rules of thumb” that promotes the creation of convenient structures. Recruitment and reinforcement such as trail laying and following in some ant species or dances in bees can be shown as the examples of positive feedback.
b. Negative feedback counterbalances positive feedback and helps to stabilize the collective pattern. In order to avoid the saturation which might occur in terms of available foragers, food source exhaustion, crowding or competition at the food sources, a negative feedback mechanism is needed.
c. Fluctuations such as random walks, errors, random task switching among swarm individuals are vital for creativity and innovation. Randomness is often crucial for emergent structures since it enables the discovery of new solutions.

In general, self-organization requires a minimal density of mutually tolerant individuals, enabling them to make use of the results from their own activities as well as activities of others.

2) Inside a swarm, there are different tasks, which are performed simultaneously by specialized individuals. This kind of phenomenon is called division of labor. Simultaneous task performance by cooperating specialized individuals is believed to be more efficient than the sequential task performance by unspecialized individuals. Division of labour also enables the swarm to respond to changed conditions in the search space.

Two fundamental concepts for the collective performance of a swarm presented above, self-organization and division of labour are necessary and sufficient properties to obtain swarm intelligent behaviour such as distributed problem-solving systems that self-organize and adapt to the given environment.

3.3 The Artificial Bee Colony Model

The colony of artificial bees consists of three groups of bees: employed bees, onlookers and scouts. The first half of the colony consists of the employed artificial bees and the second half includes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source has been exhausted by the bees becomes a scout. (Karboga, 2005)

Each cycle of the search consists of three steps: moving the employed and onlooker bees onto the food sources, calculating their nectar amounts and determining the scout bees and directing them onto possible food sources. A food source position represents a possible solution to the problem to be optimized. The amount of nectar of a food source corresponds to the quality of the solution represented by that food source. (Karaboga & Basturk, 2008)

Onlookers are placed on the food sources by using a probability based selection process. As the nectar amount of a food source increases, the probability value with which the food source is preferred by onlookers increases, too. (Karaboga & Basturk, 2008)

Every bee colony has scouts that are the colony's explorers. The explorers do not have any guidance while looking for food. They are primarily concerned with finding any kind of food source. As a result of such behavior, the scouts are characterized by low search costs and a low average in food source quality. Occasionally, the scouts can accidentally discover rich, entirely unknown food sources. In the case of artificial bees, the artificial scouts could have the fast discovery of the group of feasible solution. In this work, one of the employed bees is selected and classified as the scout bee. The selection is controlled by a control parameter called "limit". If a solution representing a food source is not improved by a predetermined number of trials, then that food source is abandoned by its employed bee and the employed bee is converted to a scout. The number of trials for releasing a food source is equal to the value of "limit" which is an important control parameter of ABC (Karboga, 2005).

In a robust search process exploration and exploitation processes must be carried out together. In the ABC algorithm, while onlookers and employed bees carry out the exploitation process in the search space, the scouts control the exploration process. In the case of real honey bees, the recruitment rate represents a "measure" of how quickly the bee swarm locates and exploits the newly discovered food source. Artificial recruiting process could similarly represent the "measurement" of the speed with which the feasible solutions or the optimal solutions of the difficult optimization problems can be discovered. The survival and progress of the real bee swarm depends upon the rapid discovery and efficient utilization of the best food resources (Karaboga & Basturk, 2008).

Similarly the optimal solution of difficult engineering problems is connected to the relatively fast discovery of "good solutions" especially for the problems that need to be solved in short time.

3.4 The Analogy between the ABC Algorithm and Engineering Design

As mentioned before, the ABC algorithm simulates the behavior of bees in finding the best food sources within a large garden of flowers. This behavior was used in determining the best frame design within a large space of possible designs. The ABC algorithm can be linked to engineering design simply by thinking of the frames as "flowers" located in the large design space "or garden". The Artificial Bee Colony would seek to obtain the best frame "or flower" which would have the largest food amount "or quality". In the current study, the quality of a frame is measured by its corresponding cost and constraint violation. That is, the lower the cost and constraint violation, the better the design.

3.5 Artificial Bee Colony Control Parameters

The sections that follow present the main control parameters of the ABC algorithm, which control the way the algorithm searches the design space to find optimum solutions.

3.5.1 Number of Bees in a Colony (N_P)

The number of bees in the colony determines the number of solutions being simultaneously investigated. A high number of bees means covering a larger area of investigation and higher computational effort. The number of bees should be related to the complexity of the problem, that is, the more complex the problem the higher the number of bees working on it.

3.5.2 The Improvement Limit for a Solution (I_L)

This is a very sensitive parameter, which affects how "deep" a bee tries to search the vicinity of a given solution and is used to escape being stuck at local minimums. If the value of this parameter is high, it indicates that the number of tries a bee does in the vicinity of a given solution is high, which also means that the bee will be stuck for a long time before considering a certain solution as abandoned and trying a different search region.

3.5.3 Maximum Number of Iterations (I_{max})

This parameter does not affect the optimization process directly, but is used to set an upper limit for the time required for optimization. The value shall be adequate to give the ABC algorithm sufficient time to converge.

3.5.4 Variable Changing Percentage (*VCP*)

This is another important parameter in the ABC algorithm. In the original algorithm, no such parameter was present since each member was defined by a single variable (such as standard steel sections). In reinforced concrete, however, a member is defined by a series of variables that define its geometry, reinforcement size and reinforcement arrangement. Thus, we need more than one variable to be changed in the solution to derive a mutant that is somehow different than the original.

3.5.5 Number of Independent Runs (r)

The number of independent runs should be adequate in order to obtain representative results and analyze the deviation of these results.

3.6 Steps of The Artificial Bee Colony (ABC) Algorithm

Same as other swarm intelligence based algorithms, the ABC algorithm has an iterative process. By assuming the number of food sources as N_S which is equal to half of the total number of bees in a colony (since the number of employed bees is equal to that of the onlooker bees), and D as the dimension of each solution vector (total number of all variables in a solution vector), the main steps of an ABC algorithm, can be defined as follows (Hadidi & Kazemzadeh, 2010):

1) A random population of solution vectors equal to the number of food sources (X_1 , ..., X_{NS}) is initialized, where $Xi = \{x_{i1}, x_{i2}, ..., x_{iD}\}$ and each solution vector is generated using:

$$x_{ij}= x_{min\,j} + rand\,[0, 1] \,.\, (x_{maxj} - x_{minj}) \quad \text{for } j = 1, 2,\ldots, D \text{ and } i = 1, 2, \ldots, N_S, \tag{3.1}$$

Where x_{maxj} and x_{minj} respectively represent the upper and lower bounds for the dimension j. After Initialization of the population, each solution is validated and set to its corresponding step size. After that, the fitness of each food source is evaluated which corresponds to its penalized cost.

2) Each employed bee searches the neighborhood of its current food source to determine a new food source v_i using:

$$v_{ij}= x_{ij}+ \phi_{ij} (x_{ij} - x_{kj}) \tag{3.2}$$

where $k \in \{1, 2, \ldots, N_S\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indexes. It must be noted that k has to be different from i, so that a new hybrid solution can be obtained. ϕ_{ij} is a random number between [-1, 1], which further assists in the hybridization process. Parameter values produced by Eq.(3.2) which exceed their boundary values are set to their boundary values and each solution is validated and set to its corresponding step size. This step is repeated a number of times equal to the number of parameters to be changed according to the variable changing percentage, discussed in section (3.4.4).

3) After generating the new food source, the nectar amount of it will be evaluated and a greedy selection will be performed. That is, if the quality of the new food source is better than the current position, the employed bee leaves its position and moves to the new food source; in other words, If the fitness of the new food source is equal or better than that of X_i , the new food source takes the place of X_i in the population and becomes a new member.

4) First an onlooker bee selects a food source by evaluating the information received from all of the employed bees. The probability (p_i) of selecting the food source i is determined by (Karboga, 2005):

$$p_i = 0.9\,\frac{f_{min}}{f_i} + 0.1 \tag{3.3}$$

Where f_i is the fitness value of the food source X_i . After selecting a food source, the onlooker generates a new food source using Eq.(3.2). Once the new food source is generated, it will be evaluated and a greedy selection will be applied, same as the case of employed bees.

5) If a candidate solution, represented by a food source cannot be further improved by a predetermined number of trials, the food source is considered abandoned and the employed bee associated with that food source becomes a scout. The scout randomly generates a new food source v_i using:

$$v_{ij} = x_{minj} + rand\,[0, 1]\,.\,(x_{maxj} - x_{minj}) \quad \text{for } j = 1, 2, \ldots, D \tag{3.4}$$

The abandoned food source is replaced by the randomly generated food source and validated to its corresponding step size. In the ABC algorithm, the predetermined number of trials for abandoning a food source was discussed in section (3.4.2), also in this algorithm at most one employed bee at each cycle can become a scout.

6) If a termination condition is met, the process is stopped and the best food source is reported; otherwise the algorithm returns to step 2. The algorithm is stopped if the maximum number if iterations I_{max} is reached, or if the algorithm does not seem to converge in its initial phase. A test run is declared divergent if it cannot obtain a correct solution, which has no penalty, within 10% of the total number of iterations. If such condition occurs, it would indicate that it isn't feasible to continue this run due to its low convergence rate.

3.7 Concluding Remarks

This chapter discussed the basic concepts that are involved in the formulation of the Artificial Bee Colony algorithm as well as the main control factors and steps that are considered in this algorithm. The information presented in this chapter is considered the basis upon which the program has been developed.

CHAPTER 4: DESIGN OF REINFORCED CONCRETE FRAMES

4.1 Introduction

Reinforced concrete frames (sometimes referred as moment resisting frame, or frames for short) are a structural system that is consisted of girders, or beams, rigidly connected with columns and are used to carry both gravity and lateral loads. This system depends on the development of rigid connections which requires proper reinforcement detailing and monolithic behavior. Usually, the design and detailing of frames is much more complicated than its shear wall counterpart, and codes of practice such as the ACI318M-08 require different degrees of ductility to be achieved in a frame depending on its seismic design category.

4.2 Reinforced Concrete Frame as a Structural System

Reinforced concrete frames, like any other structural system, have their advantages and disadvantages, as well as their own classification, which will be discussed in the following sections.

4.2.1 Advantages and Disadvantages of Reinforced Concrete Frames

The rigid nature of the beam to column joint is the key to all the advantages, as well as the reason of all disadvantages in frames. The Advantages of frames are mainly:

- Frames are capable of carrying both gravity and lateral loads efficiently.
- The rigidity of joints induces negative moments that reduce the positive span moments in the beams.
- Smaller deflection due to rotation resistance provided by frame columns stiffness.
- Frames can be designed with different degrees of ductility.
- Frames can fit in nearly any architectural planning and does not block any open spaces.

The disadvantages, however, are mainly:

- Frames with higher ductility levels require high calculation effort.
- Stability problems can arise if the inter-story sway is high.
- Frames require special care in the detailing process of joints to ensure rigid behavior.
- Although being strong in their plane of bending, frames are weak on the perpendicular plane.

4.2.2 Classification of Reinforced Concrete Frames

According to the ACI318 code, frames can be generally classified according to either their ductility requirements or their lateral deflection characteristics. Each subclass has its effect on the design process in terms of reinforcement detailing and column classification.

Frames can be classified according to their lateral deflection characteristics into (Wight & MacGregor, 2008):

1) Non-sway frames (Braced frames): Figure (4.1) shows a frame braced against sidesway in a given lateral direction. The bracing can be provided by walls, braces or buttresses designed to resist all lateral forces in that direction.

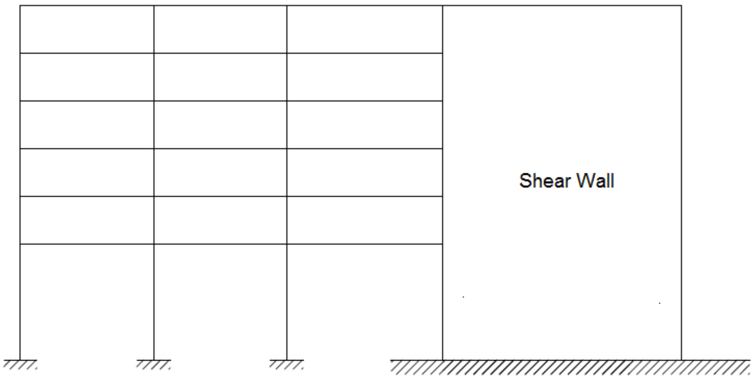


Figure 4. 1: Nonsway frame

2) Sway frames (Unbraced frames): Figure (4.2) shows an unbraced frame in which all resistance to lateral loads comes from bending in columns.

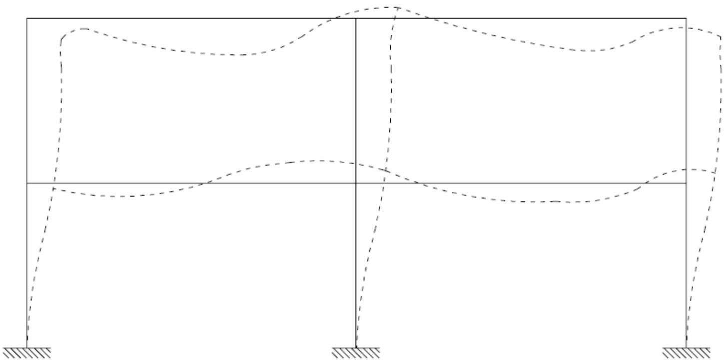


Figure 4. 2: Sway frame

There is a major difference between behavior of columns in non-sway braced frames and those in sway unbraced frames. In non-sway frames, each column acts by itself, whereas in sway frames, a column will probably not buckle individually but will probably buckle simultaneously with all other columns on the same level. As a result, it is necessary to consider the buckling length of all columns in the story (McCormac & Nelson, 2006).

Frames can also be classified according to their ductility requirements into:

1) Ordinary Moment Resisting Frames: a structural system that is designed and detailed to sustain weak earthquakes in low seismic risk zones. No special focus is required on its ductility (Hassoun & Al-Manaseer, 2008). Those frames shall satisfy the additional provisions given in ACI Code, section 21.2. Since the scope of this research deals with the design optimization of Ordinary Moment Resisting Frames, the additional provisions are listed below:
   a. ACI 21.2.2: Beams shall have at least two of the longitudinal bars continuous along both the top and bottom faces. These bars shall be developed at the face of the support.
   b. ACI 21.2.3: Columns having a clear height less than or equal five times its length, shall be designed for shear in accordance with ACI 21.3.3
2) Intermediate Moment Resisting Frames: a structural system that is designed and detailed to sustain intermediate earthquakes in intermediate seismic risk zones. Intermediate ductility level is required for those frames. Those Frames shall satisfy the additional provisions given in the ACI Code, sections 21.3.
3) Special Moment Resisting Frames: a structural system that is designed and detailed to sustain strong earthquakes in high seismic risk zones. A special focus on its ductility is required. Those frames shall satisfy the additional provisions given in the ACI Code, sections 21.5 through 21.8.

The requirements to be satisfied for different frame members at different Seismic Design categories are compiled in table (4.1).

Table 4. 1: ACI Requirements for different types of frames

Components resisting earthquake effect, unless otherwise noted	Seismic Design Category		
	Ordinary Moment Resisting Frames	Intermediate Moment Resisting Frames	Special Moment Resisting Frames
Analysis and design requirements	21.1.2	21.1.2	21.1.2, 21.1.3
Materials	None	None	21.1.4-21.1.7
Frame Members	21.2	21.3	21.5,21.6 21.7,21.8

4.3 Design of Reinforced Concrete Frame Elements

This section discusses the design of reinforced concrete frame members conforming to the ACI Code provisions. The design issues considered are only those related to the research works and will be: shear force and bending moment for beams and axial force, bending moments and shear forces for columns. Other issues considered are joint detailing, frame displacements, as well as slenderness limits.

4.3.1 Design Concept

For any member to be safe, the reduced strength of the member should be larger than the factored applied loads. Thus,

$$\emptyset R_n \geq R_u \tag{4.1}$$

Where $\emptyset$ = Strength reduction factor, R_n = Nominal resistance of a reinforced concrete member, R_u = Ultimate applied external load.

Loading cases acting on the frames considered in this work consist of lateral joint loads and uniform distributed loads. Lateral loads can be wind loads (W) or seismic loads (E), and uniform gravity loads are consisted of dead loads (D) and live loads (L). Five loading cases for each type of lateral loads are considered as suggested in the ACI code section (9.2). These cases are as follows:

1) For dead load (D), live load (L) and wind load (W)
   - **a.** U = 1.2 D + 1.6 L (4.2.a)
   - **b.** U = 1.2 D + 1.0 L + 1.6 W (4.2.b)
   - **c.** U = 1.2 D + 1.0 L – 1.6 W (4.2.c)
   - **d.** U = 0.9 D + 1.6 W (4.2.d)
   - **e.** U = 0.9 D – 1.6 W (4.2.e)
2) For dead load (D), live load (L) and Earthquake load (E)
   - **a.** U = 1.2 D + 1.6 L (4.3.a)
   - **b.** U = 1.2 D + 1.0 L + 1.4 E (4.3.b)
   - **c.** U = 1.2 D + 1.0 L – 1.4 E (4.3.c)
   - **d.** U = 1.2 D + 1.0 L + 1.4 E (4.3.d)
   - **e.** U = 1.2 D + 1.0 L – 1.4 E (4.3.e)

4.3.2 Frame Analysis

All frames are analyzed using a static first order linear analysis. According to the ACI Code (10.10.4.1), the moments of inertia for both beams and columns shall be reduced to predict the behavior of the structure at ultimate loads prior to failure. Thus the ACI Code (10.10.4.1) gives values to be used for both elastic modulus and moment of inertia. These values are:

1) Modulus of Elasticity (E_c), (ACI 8.5.1)

$$E_c = 4700\sqrt{f_c'} \quad (4.4)$$

Where E_c = Modulus of Elasticity (MPa), f_c' = Specified compressive strength of concrete (MPa).

2) Moments of Inertia

a. Columns

$$I = 0.70\, I_g \quad (4.5)$$

Where I = Moment of inertia (mm^4), I_g = Moment of inertia of gross cross section (mm^4)

b. Beams

$$I = 0.35\, I_g \quad (4.6)$$

Where I = Moment of inertia (mm^4), I_g = Moment of inertia of gross cross section (mm^4)

4.3.2.1 Relative Drift Limits

No provisions for maximum permissible inter story drift is specified in the ACI code. Other building codes such as the Uniform Building Code (UBC-97), section (1630.10.2), states that the maximum inter story drift shall not exceed 0.025 times the story height for structures having a fundamental period less than 0.7 seconds, and shall not exceed 0.020 times the story height for structures having a fundamental period larger than 0.7 seconds. Since the lateral loads applied on the frames in this research are either assumed or taken from previous study cases with no clear calculations, the upper limit for story drift is taken to be the more conservative side, thus:

$$\Delta_{rel.max} = 0.020\, H \quad (4.7)$$

Where $\Delta_{rel.max}$ = Maximum permissible relative drift between two adjacent Stories (m), H = Story Height (m)

4.3.2.2 Sway and Non-Sway Frames

Frames can be classified as either sway or non-sway. This classification has a major impact on column design by influencing its classification whether short or long, and how to magnify the moments if the column is long.

Frame classification depends on how the frame is braced and if the bracing provides lateral stiffness against sway deflections. A non-sway frame is one which has little or no lateral displacements, and thus, second order effects are limited. The ACI code gives two methods for determining whether a frame is considered sway or non-sway. The first method can be found in ACI section (10.10.5.1), which states that a column in a structure is non-sway if the increase in column end moments due to second-order effects does not exceed 5 percent of the first order end moment. The second method can be found in ACI section (10.10.5.2), which states that if the value of the Stability Index, Q, is not larger than 0.05, then the code states that the frame is considered non-sway. The classification of frames in the current study is based on ACI section (10.10.5.2), which requires Q to be calculated as follows:

$$Q = \frac{\sum P_u \Delta_0}{V_u l_c} \tag{4.8}$$

Where Q = Stability Index, $\sum P_u$= Total factored vertical load for all of the columns on the story in question (kN), Δ_0= the elastically determined first order lateral deflection due to V_u at the top of the story in question with respect to the bottom of that story (mm), V_u = the total factored horizontal shear for the story in question (kN), l_c = the height of a compression member in the frame measured from center to center of the frame joints (mm).

4.3.3 Beam Analysis and Design

Beams are horizontal members that are used usually to resist gravity loads. The main loads beams are subjected to are bending moments and shear forces. Other forces could be torsional moments and axial loads. The next sections will discuss the issue of flexure and shear only since these are used in this study.

4.3.3.1 Flexural Analysis and Design of Beams

Gravity loads applied on horizontal beams result in bending moments, which induce compression and tension stresses in a reinforced concrete beam section. The couple of forces create a couple moment that is responsible of resisting the bending moment applied on this section.

Since concrete is weak in tension, it is assumed that all tensile stresses are resisted by the reinforcement. The compression stresses, on the other hand, are resisted by concrete and reinforcements that are located in the compressive zone.

For the analysis of beams in flexure, the following assumptions are made by the ACI code section (10.2), which are as follows:

- Strain in reinforcement and concrete shall be assumed directly proportional to the distance from the neutral axis.

- Maximum usable strain at extreme concrete compression zone shall be assumed to be 0.003
- Stress in reinforcement below f_y (yield stress) shall be taken as E_s (Elasticity of Steel) times steel strain. For strains larger than yielding, yield stress shall be assumed.
- Tensile strength of concrete shall be neglected in axial and flexural calculations of reinforced concrete.
- The relationship between concrete compressive stress and concrete strain shall be assumed to be rectangular, trapezoidal, parabolic or any shape that results in an accurate prediction of strength

Based on these assumptions, and the principles mentioned before, a section under bending moments will produce two forces, as shown in Figure (4.3).

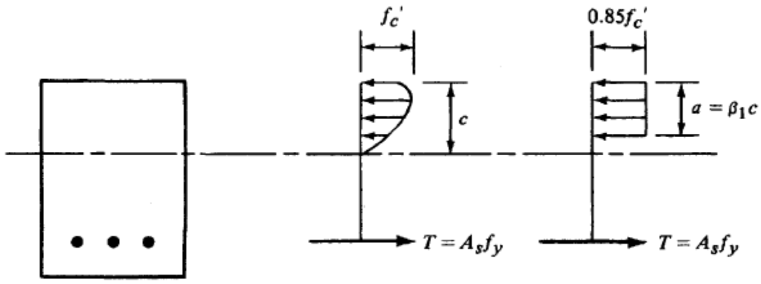


Figure 4. 3: Forces in beams under flexure and equivalent compression block

The actual shape of the compression block is parabolic, which is cumbersome to calculate and deal with, thus this parabolic stress distribution is replaced by an equivalent rectangular compression block called (Whitney's Rectangular Stress Distribution) shown in Figure (4.3) (Shihada, 2011). This rectangular stress block has an intensity of $0.85\ f_c'$ and a depth of a, which is related to the depth of the neutral axis c according to ACI section (10.2.7.3) as follows:

- For f_c' higher than 28 MPa

$$\beta_1 = 0.85 - 0.05\,\frac{(f_c' - 28)}{7} \geq 0.65 \quad (4.9.a)$$

- For f_c' between 17 MPa and 28 MPa

$$\beta_1 = 0.85 \quad (4.9.b)$$

Thus,

$$a = \beta_1 c \quad (4.10)$$

Referring to Figure(4.4) and Figure(4.5), from simple statics we can derive the equations that govern the flexural strength of beams:

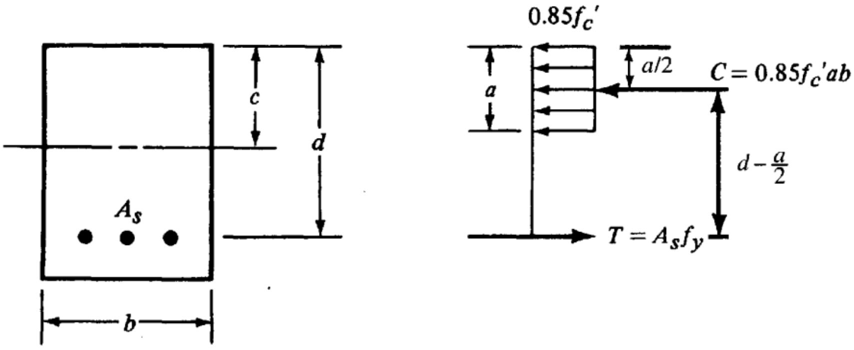


Figure 4. 4: Equilibrium of forces in beams

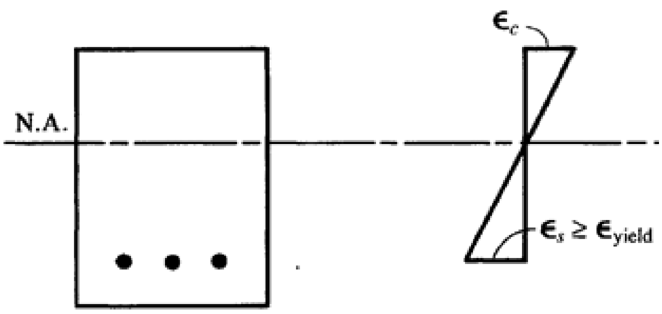


Figure 4. 5: Strain compatibility in beams

For equilibrium of forces,

$$C = T \quad (4.11)$$

$$0.85\, f_c'ab = A_sf_y \quad (4.12)$$

Thus,

$$a = \frac{A_sf_y}{0.85\, f_c'b} \quad (4.13)$$

Where C = Compressive force resulting from compressive stress block (N), T = Tension force resulting from reinforcement (N), fc' = Specified compressive strength of concrete (MPa), a = depth of Whitney's rectangular compression block (mm), b = width of beam (mm), f_y = yielding strength of reinforcement (MPa).

And for equilibrium of moments,

$$\emptyset M_n = \emptyset A_s f_y (d - \frac{a}{2}) \quad (4.14)$$

Substituting for (a) from Eq.(4.13), we obtain,

$$\emptyset M_n = \emptyset A_s f_y \left(d - \frac{A_s f_y}{1.7 \, f_c' b} \right) * 10^{-6} \quad (4.15)$$

Where $\emptyset$ = Strength reduction factor calculated according to ACI section (9.3.2.1) and (9.3.2.2), M_n = Nominal flexure capacity of beam (kN.m), A_s = Total area of tensile reinforcement (mm^2), d = distance from the center of tension reinforcement to the extreme compression fiber (mm).

According to ACI section (9.3.2.1) and (9.3.2.2), cross sections can be classified as Compression Controlled, Tension Controlled and Transition Sections. ACI states that for compression controlled sections having a net tensile strain in extreme tension steel equal to or smaller than 0.002 when concrete reaches its crushing strain of 0.003, the strength reduction factor $\emptyset$ is taken as 0.65. Whereas for tension controlled sections having a net tensile strain in extreme tension steel equal to or larger than 0.005 when concrete reaches its crushing strain of 0.003, the strength reduction factor $\emptyset$ is taken as 0.9. Sections between these two extremes are called transition sections and the strength reduction factor $\emptyset$ is calculated by linear interpolation, thus $\emptyset$ can be related to extreme tension strain ϵ_t as shown

$$\emptyset = 0.65 + (\epsilon_t - 0.002)(250/3) \;\; and \; 0.65 \leq \emptyset \; \leq 0.9 \quad (4.16)$$

According to ACI section (10.3.5), for members with axial force less than or equal to $0.1 f_c' A_g$, the strain of extreme reinforcement ϵ_t shall not be less than 0.004.

Eq.(4.15) is used to determine the flexural strength of a reinforced concrete beam section of known dimensions and reinforcement, as well as designing reinforced concrete beam sections under flexural moments.

4.3.3.2 Development of Flexural Reinforcement

In addition to creating stresses in reinforcement and concrete, flexural deformations of a beam also create stresses between the reinforcement and concrete called bond stresses. If the intensity of these stresses is not restricted, they may produce crushing or splitting of the concrete surrounding the reinforcement, especially if bars are closely spaced or located near the surface of the concrete. Failure of the concrete permits the reinforcement to slip. As slipping occurs, the stress in the reinforcement drops to zero, and the beam which behaves as if it was made of plain concrete is subject to immediate failure as soon as the concrete cracks (Shihada, 2011). According to ACI section 12.2.3, the development length (l_d) is given by,

$$l_d = \left(\frac{f_y}{1.1\lambda\sqrt{f_c'}} \frac{\psi_t \psi_e \psi_s}{\left(\frac{c_b + K_{tr}}{d_b} \right)} \right) d_b \tag{4.17}$$

Where ψ_t = factor used to modify development length based on reinforcement location, ψ_e = factor used to modify development length based on reinforcement coating, ψ_s = factor used to modify development length based on reinforcement size. c_b = smaller of distance from center of bar to concrete surface, and half of center to center spacing between bars (mm). K_{tr} = transverse reinforcement index, d_b = diameter of reinforcement bar being developed (mm).

The factors used in the expressions for development of deformed bars in tension are as follows:

1) Where horizontal reinforcement is placed such that more than 300mm of fresh concrete is cast below the development length or splice, ψ_t = 1.3. For other situations ψ_t = 1.0.
2) For epoxy-coated bars with cover less than $3d_b$, or clear spacing less than $6d_b$, ψ_e = 1.5. For all other epoxy-coated bars, ψ_e = 1.2. For other cases, ψ_e = 1.0.
3) For No. 19 and smaller bars, ψ_s = 0.8. For No. 22 and larger bars, ψ_s = 1.0.
4) Where lightweight concrete is used, λ shall not exceed 0.75. Where normalweight concrete is used, λ = 1.0

The confinement term $\left(\frac{c_b+K_{tr}}{d_b}\right)$ shall not exceed 2.5 and,

$$K_{tr} = \frac{40A_{tr}}{sn} \tag{4.18}$$

Where A_{tr} = area of transverse reinforcement (mm^2), s = maximum center-to-center spacing of transverse reinforcement within development length l_d (mm), n = number of bars being developed along plane of splitting.

According to ACI section (12.2.3), K_{tr} is permitted to be taken as K_{tr} = 0 as a design simplification even if transverse reinforcement is present.

4.3.3.3 Bar Cutoffs

Some of the flexural reinforcement bars can be cutoff where they are no longer needed to resist tensile forces or where the remaining bars are adequate to do so. In a continuous beam of constant cross section, if the areas of steel required at the sections of maximum moment are made continuous throughout each region of positive or negative moment, the beam will be over-designed at most sections. It is often desirable to terminate a portion of the steel when the moment decreases significantly. Reducing the area of reinforcement in regions of low bending moment in a concrete element lowers the cost of the element. Furthermore, for heavily reinforced elements, the reduction in a number of reinforcement bars improves concrete casting and compaction operations (Shihada, 2011).

There must be sufficient extension of each bar, on each side of every critical moment section to develop the force in that bar at that section. Tension bars, cutoff in a region of moderate shear force, cause a major stress concentration which can lead to major inclined cracks at the bar cutoff. Thus, bar cutoffs should be kept to a minimum, particularly, in zones of tension for ease of design and fabrication.

Several considerations shall be taken into account, these considerations are stated in the ACI code section (12.10), shown in Figure(4.6) and are summarized as follows:

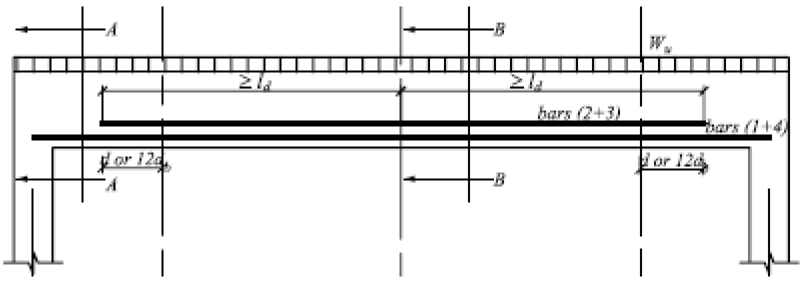


Figure 4. 6: Considerations for bar cut off (Shihada, 2011)

- According to ACI Code 12.10.2, critical sections for development of reinforcement in flexural members are at points of maximum stress and at points within the span where adjacent reinforcement terminates, or is bent.
- To account for the possibility of higher than anticipated moment at cutoff point due to possible variations in the position of live load, settlements of support, lateral loads, or other causes, ACI Code 12.10.3 requires the reinforcement to be extended beyond the point at which it is no longer required to resist flexure for a distance equal to the effective depth of the member d or $12d_b$, whichever is greater, except at supports of simple spans and at free ends of cantilevers. When bars of different sizes are used, the extension should be in accordance with the diameter of bar being terminated.
- Based on ACI Code 12.10.4, continuing reinforcement is to have an anchorage length not less than the development length (l_d) beyond the point where bent or cutoff reinforcement is no longer required to resist flexure.
- Flexural cracks tend to open early whenever any reinforcement is cutoff in a tension zone. If the steel stress in the continuing reinforcement and the shear strength are near their ultimate values, diagonal tension cracking tend to develop too early from these

flexural cracks. Diagonal cracks are less likely to form where shear stress is low. Diagonal cracks can be restrained by closely spaced stirrups.

4.3.3.4 Shear Analysis and Design

As mentioned before, loads applied on beams produce shear forces and bending moments that a reinforced concrete beam needs to resist. The analysis and design of reinforced concrete beams to bending moments was considered in the previous section. In this section, the design of reinforced concrete to resist shear is dealt with.

ACI section (11.1.1) gives the nominal shear strength of a reinforced concrete beam section as,

$$V_n = V_c + V_s \tag{4.19}$$

Considering the strength reduction factor $\emptyset$, Eq.(4.19) becomes,

$$\emptyset V_n = \emptyset (V_c + V_s) \tag{4.20}$$

Where $\emptyset$ = Strength reduction factor and equals (0.75) for shear as per ACI (9.3.2.3), V_n = nominal shear strength of a reinforced concrete beam section (kN), V_c = nominal shear strength provided by concrete (kN), V_s = nominal strength provided by shear reinforcement (kN)

The ACI code gives numerous formulae for the calculation of the nominal strength provided by concrete. The simplest among these is the formula stated in ACI section (11.2.1.1), which is used in this research and is given by,

$$V_c = 0.17\,\lambda \sqrt{f_c'}\, b_w d * 10^{-3} \tag{4.21}$$

Where λ = modification factor reflecting the reduced mechanical properties of lightweight concrete (λ =1 for normal weight concrete), b_w = width of beam web (mm).

The strength of shear reinforcement perpendicular to the axis of the reinforced concrete beam is given by ACI section (11.4.7.2) as,

$$V_s = \frac{A_v f_{yt} d}{s} * 10^{-3} \tag{4.22}$$

Where s = center to center spacing between shear reinforcement ties (mm), A_v = area of transverse reinforcement (mm^2), f_{yt} = yield strength of reinforcement (MPa).

4.3.3.5 Deflection Control

Two methods are given in the ACI Code for controlling deflections for beams not supporting or attached to partitions or other construction likely to be damaged by large deflections. The first method indirectly controls deflection by means of minimum thickness to a ratio of the span length (L), as shown in Table (4.2) and stated in ACI section (9.5), and the second by directly limiting computed deflections, as shown in Table (4.3).

Table 4. 2: Minimum thickness for reinforced concrete beams

Restraint	Simply Supported	One End Continuous	Both Ends Continuous	Cantilever
Minimum thickness for beams*	$L/16$	$L/18.5$	$L/21$	$L/8$

* Valid for f_y = 420 MPa, for other f_y, the value is modified by $(0.4 + f_y/700)$

Table 4. 3: Maximum permissible computed deflections

Type of member	Deflection to be considered	Deflection limitation
Flat roofs not supporting or attached to non-structural elements likely to be damaged by large deflections	Immediate deflection due to live load L	$L/180$
Floors not supporting or attached to non-structural elements likely to be damaged by large deflections	Immediate deflection due to live load L	$L/360$
Roof or floor construction supporting or attached to non-structural elements likely to be damaged by large deflections	That part of the total deflection occurring after attachment of nonstructural elements (sum of the long-term deflection due to all sustained load and the immediate deflection due to any additional live load	$L/480$
Roof or floor construction supporting or attached to non-structural elements not likely to be damaged by large deflections		$L/240$

The deflection characteristics of reinforced concrete frames is much better than those of their normal beam counter parts (those which are not part of a moment resisting frame), thus, one can use the limits stated in Table (4.2) as a conservative simplification. Such a simplification was used by (Akin & Saka, 2011).

4.3.3.6 Other Design Considerations

When designing reinforced concrete beams, ACI gives certain constraints and provisions to ensure ductility and safety of beams under ultimate loads, these provisions will be covered in section (4.4.3).

4.3.4 Column Analysis and Design under Axial loads and Bending

All columns are subjected to some bending as well as axial forces, and they need to be proportioned to resist both (McCormac & Nelson, 2006). Columns will bend under the action of moments, and those moments will tend to produce compression on one side of the column and tension on the other.

4.3.4.1 Strength Interaction Diagrams

Columns are required to resist both axial forces and bending moments that are not independent of each other. The interdependency of axial forces and bending moment result in a so-called interaction diagram that gives the combination of axial force and bending moment that results in the failure of a reinforced concrete sections (Wight & MacGregor, 2008).

Figure(4.7) illustrates such an interaction diagram with 5 characteristic points with their strain distributions shown.

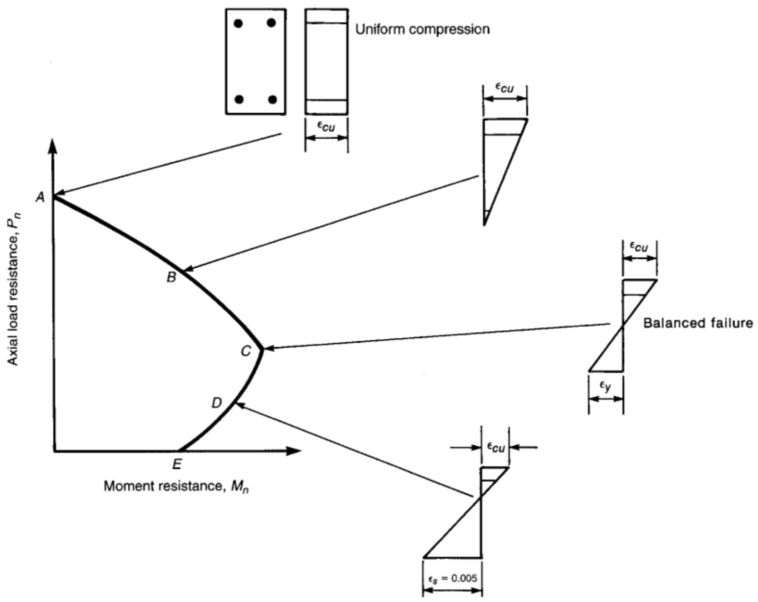


Figure 4. 7: Nominal interaction diagram

These key points and regions of the interaction diagram are discussed below:

1) Point A – Pure Axial Load: Point A in Figure(4.7) and its corresponding strain distribution represent uniform axial compression without moment. This is the largest nominal axial load the column can support.
2) Point B – Zero Tension, Onset of Cracking: The strain distribution at B in Figure(4.7) corresponds to the axial load and moment at the onset of crushing of the concrete just as the strains in the concrete on the opposite face of the column reaches zero. Case B represents the onset of cracking of the least compressed side of the column. Because tensile stresses are ignored in the strength calculations, failure loads below point B in the interaction diagram represent cases where the section is partially cracked .
3) Region A – C – Compression Controlled Failures: Column with axial loads P_u and moments M_u that fall on the upper branch of the interaction diagram between points A and C initially fail due to crushing of the compression face before extreme tensile layer of reinforcement yields. Hence they are called compression controlled columns.

4) Point C – Balanced Failure, Compression Controlled Limit Strain: Point C in Figure(4.7) corresponds to a strain distribution with a maximum compressive strain of 0.003 on one face of the section and tensile strain equal to the yield strain in the layer of reinforcement farthest from the compression face of the column.
5) Point D – Tensile Controlled Limit: Point D in Figure(4.7) corresponds to a strain distribution with 0.003 compressive strain on the top face and tensile strain of 0.005 in the extreme layer of tension steel. The failure of such column will be ductile, with steel strains at failure that are about two and a half times the yield strain for 420 MPa reinforcement. The strain of 0.005 was chosen to be significantly higher than yielding strain to ensure ductile failure.
6) Region C – D – Transition Region: Flexural members and columns with loads and moments which would plot between points C and D in Figure(4.7) are called transition failures because the magnitude of the curvatures at the critical section are in a transition between the ultimate curvature corresponding to steel strains of 0.002 and 0.005. This is reflected in the transition of the strength reduction factor Ø from 0.65 to 0.9 rectangular tied columns.

4.3.4.2 Derivation of Computation Method for Interaction Diagrams

In this section, the relationship needed to compute the various points on an interaction diagram are derived by using strain compatibility and simple statics. The calculation of an interaction diagram involves the basic assumptions stated in the ACI section (10.2). The derivation is limited to rectangular tied columns shown in Figure(4.8.a), since these are the columns used in this study.

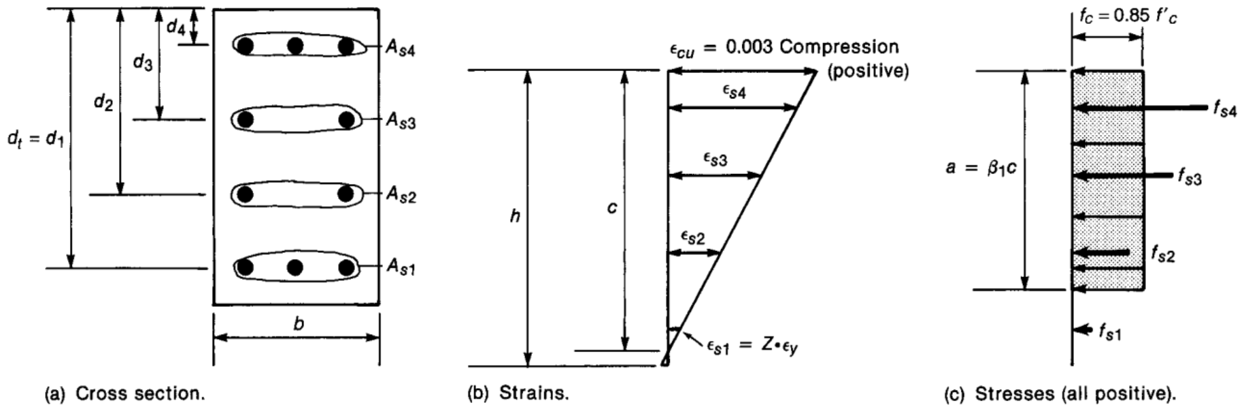


Figure 4. 8: Forces, strains and stresses in a column section (Wight & MacGregor, 2008)

The general case involves the calculation of nominal compression force P_n acting at the centroid and the nominal bending moment M_n acting about the centroid of the gross cross section, for an assumed strain distribution with concrete crushing strain ϵ_{cu} = 0.003. The column cross section and the assumed strain distribution are shown in Figure(4.8.a) and Figure(4.8.b), Four layers of reinforcement are shown, layer 1 having strain ϵ_{s1} and area A_{s1}, and so on. Layer 1 is the closest to the "least compressed" surface and is at a distance d_1 from the "most compressed" surface. Layer 1 is called the extreme tension layer. It has depth d_t and a strain ϵ_t.

The strain distribution will be defined by setting ϵ_{cu} = 0.003 and assuming a value for ϵ_{s1}. An iterative calculation will be necessary to consider a series of cases as shown in Figure (4.7). The iteration can be controlled by selecting a series of values for the neutral axis depth, *c*. Large values of *c* will give points high in the interaction diagram while low values of *c* will give points low in the interaction diagram. To find points corresponding to specific values of strain in the extreme layer of tension reinforcement, the iteration can be controlled by setting $\epsilon_{s1} = Z\,\epsilon_y$, where *Z* is an arbitrarily chosen value. Positive values of *Z* correspond to positive (compressive) strains as shown in Figure (4.8.b). For example, *Z* = -1 corresponds to $\epsilon_{s1} = -\epsilon_y$, the yield strain in tension. Such a strain distribution corresponds to the balanced-failure condition.

Now with reference to Figure (4.8. a,b and c), and by similar triangles,

$$c = \left(\frac{0.003}{0.003 - Z\epsilon_y}\right) d_1 \tag{4.23}$$

and,

$$\epsilon_{si} = \left(\frac{c - d_i}{c}\right) 0.003 \tag{4.24}$$

Where ϵ_{si}= the strain in the *i*th layer of steel, d_i = the depth of the i^{th} layer of steel.

Once strains are known, we can use hook's law to determine the stress in the reinforcement bars, taking into account that stresses beyond yielding are set to be the yielding stress,

$$f_{si} = \epsilon_{si}\, E_s \quad \text{but} \quad -f_y \le f_{si} \le f_y \tag{4.25}$$

Where E_s = the elastic modulus of reinforcing steel bars. (MPa)

The stresses in concrete are represented by whitney's rectangular stress block discussed before, thus,

$$a = \beta_1 c \tag{4.26}$$

Where the value of β_1 is the same as discussed previously,

- For fc' higher than 28 MPa

$$\beta_1 = 0.85 - 0.05\,\frac{(f_c' - 28)}{7} \geq 0.65 \qquad (4.27.a)$$

- For fc' between 17 MPa and 28 MPa

$$\beta_1 = 0.85 \qquad (4.27.b)$$

After establishing stresses and strains, the next step is to compute the compressive force in the concrete, C_c, and the forces in each layer of reinforcement, F_{s1}, F_{s2}, and so on. This is done by multiplying the stresses by their corresponding areas. Thus,

$$C_c = (0.85 f_c')(ab) \qquad (4.28)$$

And for reinforcements, if a is less than d_i,

$$F_{si} = f_{si} A_{si} \quad \text{(Positive in compression)} \qquad (4.29.a)$$

If a is greater than d_i, the area of reinforcement in that layer has already been included in the area (ab) that is used to compute C_c. As a result, it is necessary to substract $0.85f_c'$ from f_{si} before computing F_{si}, thus,

$$F_{si} = (f_{si} - 0.85 f_c') A_{si} \qquad (4.29.b)$$

The resulting forces of concrete and reinforcements are shown in Figure (4.9),

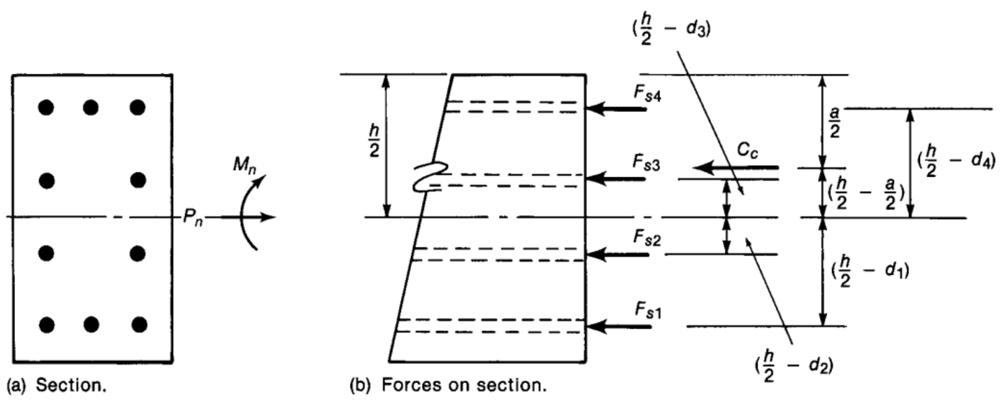


Figure 4. 9: Cross section and forces on reinforcement layers

The nominal axial load capacity, P_n, for the assumed strain distribution is the summation of the axial forces, thus,

$$P_n = C_c + \sum_{i=1}^{n} F_{si} \tag{4.30}$$

The nominal moment capacity M_n, for the assumed strain distribution is found by summing the moments of all the internal forces about the centroid of the column. The moments are summed about the centroid of the section, because this is the axis about which moments are computed in conventional structural analysis, thus for rectangular section,

$$M_n = C_c\left(\frac{h}{2} - \frac{a}{2}\right) + \sum_{i=1}^{n} F_{si}\left(\frac{h}{2} - d_i\right) \tag{4.31}$$

The values of M_n and P_n are plotted to result in a "nominal interaction diagram". To derive the "strength interaction diagram" two modifications have to be done to the nominal interaction diagram:

1) The values of M_n and P_n are multiplied by the strength reduction factor $\emptyset$, which has a value between 0.65 and 0.9 depending on the strain of the extreme tension layer
2) The maximum compressive force is limited to the uniaxial compression capacity of columns which is:

$$\emptyset P_n = \emptyset\emptyset'\left(0.85\, f_c' A_g + A_s(f_y - 0.85\, f_c')\right) \tag{4.32}$$

Where $\emptyset$ = Strength reduction factor ($\emptyset$=0.65), $\emptyset'$= Reduction factor that accounts for accidental eccentricities ($\emptyset'$=0.8), A_g = gross area of cross section (mm^2), A_s = Reinforcement area (mm^2), f_y = yielding strength of reinforcement (MPa), f_c'= Specified compressive strength of concrete.

To finalize, Figure(4.10) shows both nominal and strength interaction diagrams for a column, taken from (Wight & MacGregor, 2008).

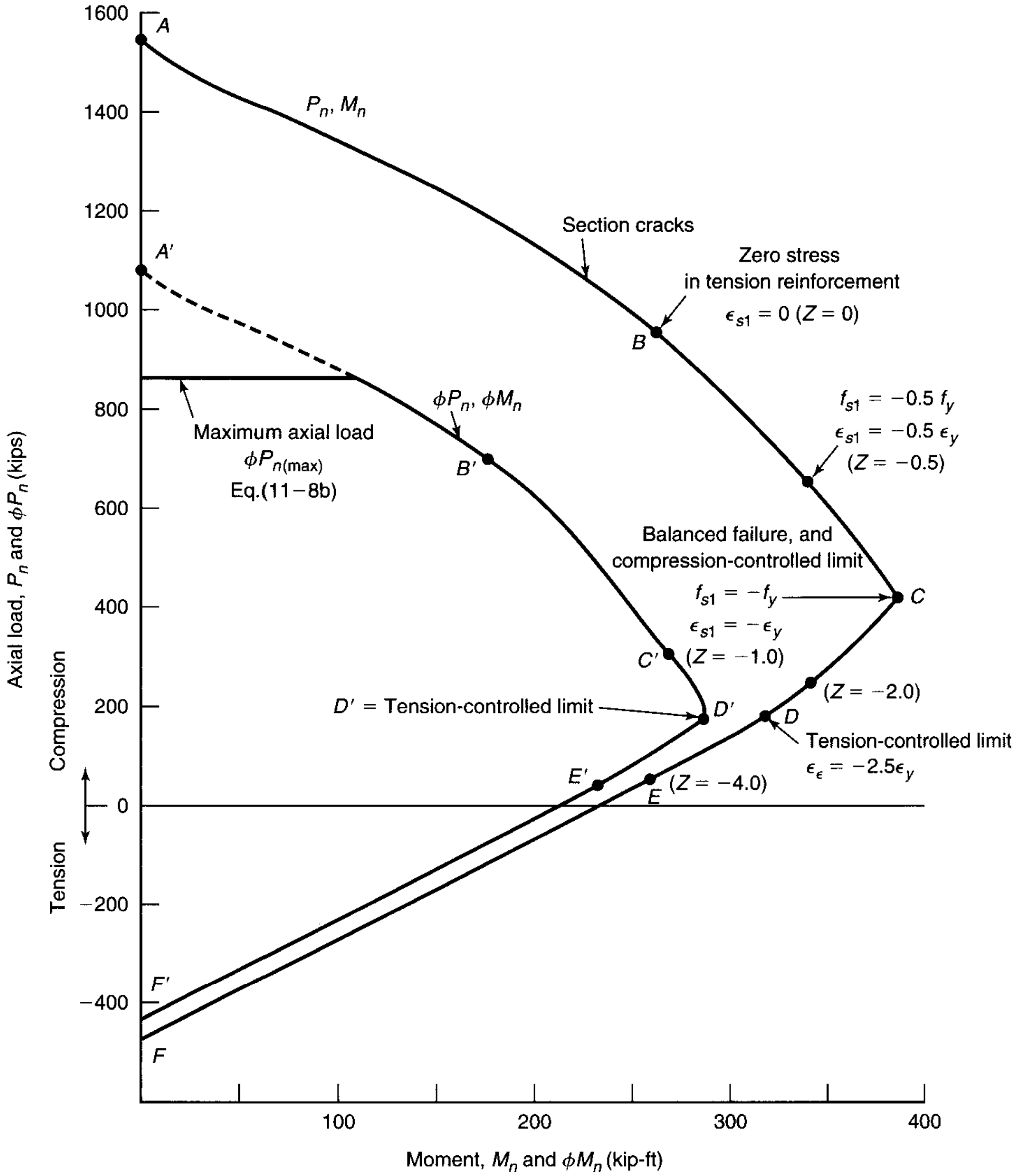


Figure 4. 10: Nominal and strength interaction diagrams (Wight & MacGregor, 2008)

4.3.4.3 Slender columns

When a column bends or deflects laterally an amount Δ, its axial load will cause an increased column moment equal to PΔ. This moment will be superimposed onto any moments already in the column. Should this PΔ moment be of such magnitude as to reduce the axial load capacity of the column significantly, the column will be referred to as a slender column (McCormac & Nelson, 2006).

According to ACI section (10.10.1), slenderness effect shall be permitted to be neglected in the following cases:

1) For compression members not braced against sidesway when:

$$\frac{kl_u}{r} \leq 22 \tag{4.33}$$

2) For compression members braced against sidesway when:

$$\frac{kl_u}{r} \leq 34 - 12\frac{M_1}{M_2} \leq 40 \tag{4.34}$$

Where k = effective length factor, l_u =unbraced length of column (m), r = radius of gyration (m), M_1 = smaller end moment on column, which is positive if column bent into single curvature and negative if bent into double curvature(kN.m), M_2 = larger end moment on column (kN.m).

The effective length factor is obtained by computing the relative stiffness of columns to beams at both column ends, then using the appropriate Jackson Moreland alignment chart, the effective length factor k can be obtained. Another alternative method is to use equations that obtain k for a given relative stiffness, which best suits computer programming and is used in this study.

Thus, to find k, the relative stiffness of columns to beams must be obtained according to ACI section (10.10.1) as follows,

$$\psi = \frac{\sum(E_c I_c / l_c)}{\sum(E_b I_b / l_b)} \tag{4.35}$$

Where ψ = relative stiffness, E_c = Modulus of Elasticity of columns (MPa), I_c = Moment of Inertia of columns (m^4), l_c = Length of columns (m), E_b = Modulus of Elasticity of beams (MPa), I_b = Moment of Inertia of beams (m^4), l_b = Length of beams (m).

According to ACI section (10.10.4.1), the moment of inertia for beams and columns shall be reduced to account for near ultimate behavior of the structural members in a frame, which result in an overestimation of the second-order deflections on the order of 20 to 25 percent. These reduction factors were stated in section (4.3.2).

It should be noted that for supports being assumed as pin, the relative stiffness according to Eq.(4.35) goes to infinity, since the assumption of pinned supports is similar to a support with beams of zero stiffness. Pin supports, however, are never completely frictionless, thus a value for ψ should be taken as 10 instead of infinity. Same goes for fixed supports, ψ should be taken as 1 instead of 0 (Wight & MacGregor, 2008).

Once the relative stiffness at each end of a column has been calculated, the effective length factor can be obtained from the following equations (McCormac & Nelson, 2006):

For nonsway frames, k is the smaller value of the two following equations:

$$k = 0.7 + 0.05\ (\psi_a + \psi_b) \leq 1 \tag{4.36.a}$$

$$k = 0.85 + 0.05\psi_{min} \leq 1 \tag{4.36.b}$$

Where ψ_a = relative stiffness at bottom end, ψ_b = relative stiffness at top end, ψ_{min} = smaller of ψ_a and ψ_b

For sway frames, k can be calculated as follows:

$$for\ \psi_m < 2 \quad k = \frac{20 - \psi_m}{20}\sqrt{1 + \psi_m} \tag{4.37.a}$$

$$for\ \psi_m \geq 2 \quad k = 0.9\sqrt{1 + \psi_m} \tag{4.37.b}$$

Where ψ_m = the average value of ψ_a and ψ_b

Once the effective length factor has been determined, the coefficient $\frac{kl_u}{r}$ can be calculated, which enables the columns to be classified to long and short according to Eq. (4.33) and Eq. (4.34).

If a column is classified as long, or slender, the moment is magnified to account for the effect of slenderness on the strength of a column. The moment magnification procedure for non-sway frames is done according to the ACI code section (10.10.6), whereas ACI code section (10.10.7) covers the moment magnification for sway frames.

4.3.4.4 Moment Magnification Procedure for Non-sway Frames

According to ACI section (10.10.6), compression members that are classified as being long, or slender, and are part of a non-sway frame, shall be designed for a factored axial force (P_u) and the factored moment amplified to account for the effects of member curvature (M_c) where,

$$M_c = \delta_{ns} M_2 \tag{4.38}$$

Where M_c = magnified design moment (kN.m), δ_{ns} = non-sway moment magnification coefficient.

The moment magnification coefficient can be calculated as follows,

$$\delta_{ns} = \frac{C_m}{1 - \frac{P_u}{0.75P_c}} \geq 1.0 \tag{4.39}$$

Where C_m = factor relating actual moment diagram to an equivalent uniform moment diagram, P_u = factored axial force on column (kN), P_c = critical buckling load (kN)

The critical buckling load is determined by,

$$P_c = \frac{\pi^2 EI}{(kl_u)^2} \tag{4.40}$$

ACI section (10.10.6.1) states that the value (*EI*) shall be calculated as follows:

$$EI = \frac{0.4E_c I_g}{1+\beta_{dns}} \tag{4.41}$$

Where β_{dns} = the ratio of maximum factored axial sustained load to maximum factored axial load associated with the same load combination, but shall not be taken greater than 1.0.

The factor C_m can be determined by,

$$C_m = 0.6 + 0.4\frac{M_1}{M_2} \tag{4.42}$$

Where the ratio M_1/M_2 is positive if the column is bent in single curvature, and negative if the member is bent in double curvature. For members with transverse loads between supports C_m = 1.0.

The ACI code also states that the value of M_2 shall be larger than a certain minimum calculated according to ACI section (10.10.6.5) as follows,

$$M_{2,min} = P_u(15 + 0.03h) \tag{4.43}$$

Where *h* = cross section dimension in the plane of bending (mm).

Once the moments are amplified, the factored axial forces and amplified factored moments are used to design the column using its derived strength interaction diagram.

4.3.4.5 Moment Magnification Procedure for Sway Frames

According to ACI section (10.10.7), long column in sway frames shall be designed for a factored axial force (P_u) and the factored moment amplified to account for the effects of member curvature (M_c) where,

$$M_c = M_{2\,ns} + \delta_s M_{2s} \tag{4.44}$$

Where M_{2ns} = Factored bending moment resulting from non-sway loads (kN.m), δ_s = sway moment magnification factor, M_{2s} = Factored bending moment resulting from sway loads (kN.m).

Where δ_s is calculated according to ACI 10.10.7.3 as follows,

$$\delta_s = \frac{1}{1-Q} \geq 1.0 \tag{4.45}$$

Once amplification is done, the amplified moment and factored axial force is used to design the column using its corresponding interaction diagram.

4.3.4.6 Other Design Considerations

When designing reinforced concrete columns, ACI gives certain constraints and provisions to ensure ductility and safety of columns under ultimate loads, these provisions will be covered in section (4.4.2)

4.3.5 Joint Detailing

The design of joints requires knowledge of the forces to be transferred through the joint and the most likely ways in which this transfer can occur. The ACI code discusses joint design in several sections:

1) ACI code section (7.9) requires enclosure of splices of continuing bars and of the end anchorages of bars terminating in connections of primary framing members, such as beams and column.
2) ACI code section (11.10.2) requires a minimum amount of lateral reinforcement (ties or stirrups) in beam-column joints if the joints are not restrained on all four sides by beams or slabs of approximately equal depth. The amount required is the same as the minimum stirrup requirement for beams.
3) ACI code section (12.12.1) requires negative-moment reinforcement in frames to be anchored in, or through, the supporting member by embedment length, hooks, or mechanical anchorage.
4) ACI code section (12.11.2) requires that, in frames forming the primary lateral load-resisting system, a portion of the positive-moment steel should be anchored in the joint to develop the yield strength, f_y, in tension at the face of the support.

Recommendations for design of beam-column connections in monolithic reinforced concrete structures can be found in the ACI committee 352 (ACI 352R-02). The ACI352 gives guidelines for the design of joints based on shear as well as the detailing considerations.

According to ACI352 section (4.5.2.3), for Type 1 connections (parts of ordinary moment resisting frames), the development length of a bar terminating in a standard hook within a joint should be computed as follows:

$$l_{dh} = \frac{f_y d_b}{4.2\sqrt{f_c'}} \tag{4.46}$$

Where l_{dh} = anchorage of bars developed in joint (mm), f_y = Yield stress of steel (MPa), d_b = diameter of bar being developed (mm), f_c' = specified concrete compressive strength (MPa).

4.3.5.1 Detailing of Opening Corner Joints

If a corner joint of a rigid frame tends to be opened by the applied moments, it is called “opening joint”. Figure (4.11.a) shows the internal forces inside an opening joint, it is clear that these forces cause tensile and compressive stresses as shown in Figure (4.11.b). These stresses will cause cracking in concrete in patterns shown in Figure (4.11.c).

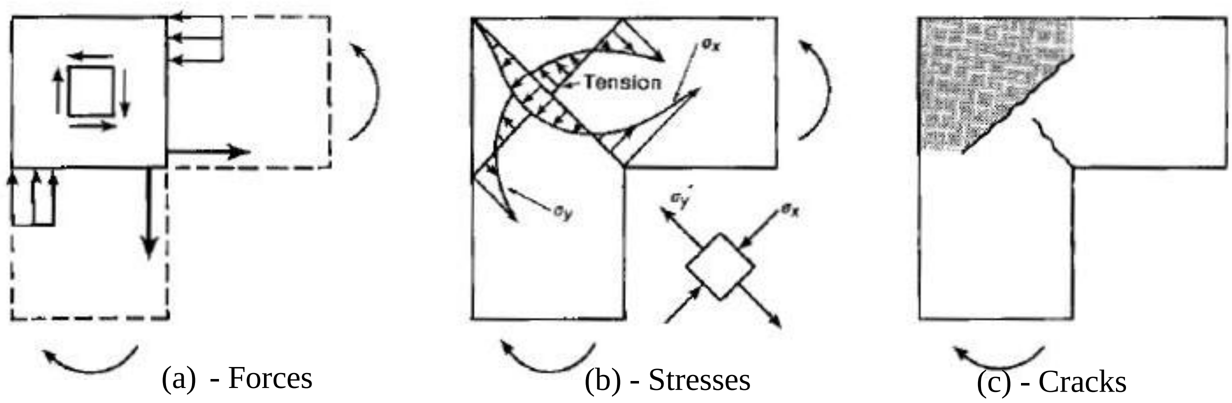


Figure 4. 11: Opening joints in frames (Wight & MacGregor, 2008)

Different detailing schemes were investigated by (Nilsson, 1973) under bending moments that tend to open the joint. These results are shown in Figure (4.12) together with the detailing that gave such results.

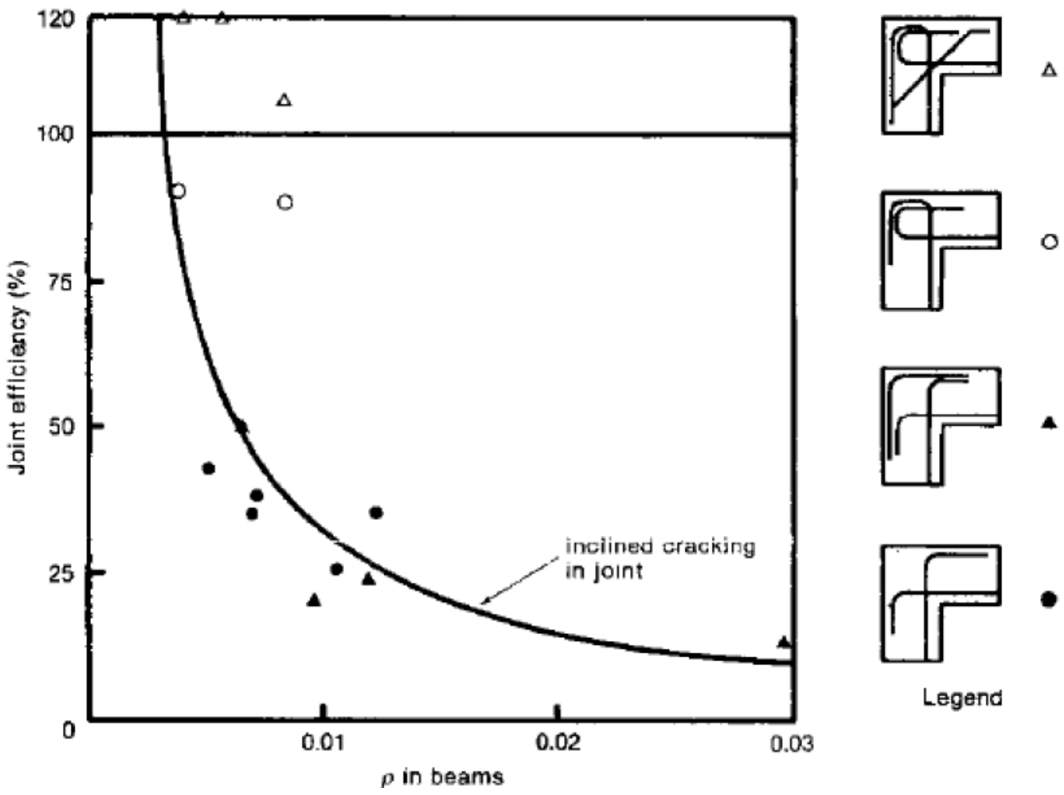


Figure 4. 12: Efficiency of different opening joint detailing schemes (Shehada, 2011)

The joint detailing used for corner joints classified as opening joints in this study is shown in Figure (4.13).

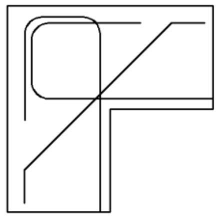
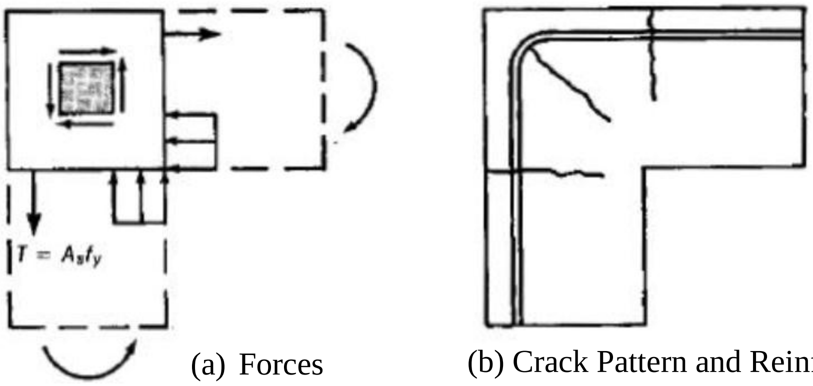


Figure 4. 13: Joint detailing scheme used for opening corner joints

4.3.5.2 Detailing of Closing Corner Joints

If a corner joint of a rigid frame tends to be closed by the applied moments, it is called “closing joint”. Figure (4.14.a) shows the tensile and compressive forces in a joint. The reinforcement detailing and crack pattern is shown in Figure (4.14.b)



(a) Forces (b) Crack Pattern and Reinforcement

Figure 4. 14: Closing joints in frames (Wight & MacGregor, 2008)

4.3.5.3 Detailing of T Joints

The detailing scheme for T-Joints used in this study is shown in Figure (4.15) (Shehada, 2011).

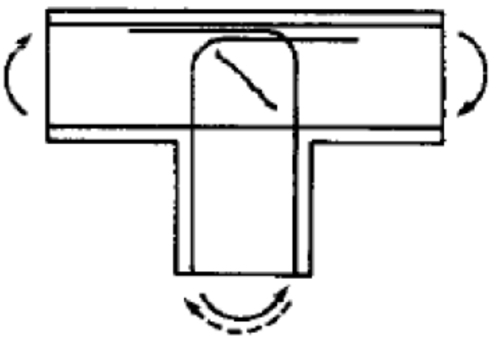


Figure 4. 15: Detailing of T-Joints

4.3.5.4 Detailing of Exterior Joints

The detailing scheme for exterior joints used this study is shown in Figure (4.16) (Shehada, 2011).

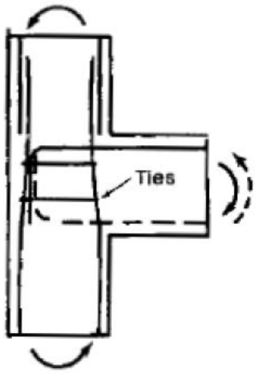


Figure 4. 16: Detailing of exterior joints

4.4 Design Constraints and Provisions

This section discusses the design constraints considered in the optimization process of reinforced concrete frames, these design constraints will be described here and mathematical expression will be derived later on in chapter(5).

4.4.1 Frame Constraints

In the overall analysis of reinforced concrete frames, relative inter story drift limit was the only constraint that was considered. The maximum permissible inter story drift is not specified in the ACI code, but can be found in other building codes such as (UBC-97), which states in section (1630.10.2) that the maximum story drift shall not exceed the following,

$$\Delta_{rel.max} = 0.020\ H \qquad (4.47)$$

Where $\Delta_{rel.max}$ = Maximum permissible relative drift between two adjacent Stories (m), H = Story Height (m)

4.4.2 Column Constraints

The following constraints were considered for the design of columns:

1) Strength constraint:

The column shall have adequate strength to resist the applied factored axial force and factored bending moment after magnification, if applicable, is done. That is,

$$\phi P_n \geq P_u \qquad (4.48)$$

and,

$$\phi M_n \geq M_u \qquad (4.49)$$

Where $\emptyset$ = Strength reduction factor, P_n = Nominal axial force resistance of a reinforced concrete column (kN), P_u = Ultimate applied axial force (kN), M_n = Nominal bending moment resistance of a reinforced concrete column (kN.m), M_u = Ultimate applied magnified bending moment (kN.m).

2) Minimum reinforcement ratio:
According to ACI section (10.9.1), the minimum reinforcement ratio shall be 1% of the gross cross sectional area. Thus,

$$A_{st,min} = 0.01\ A_g \tag{4.50}$$

Where A_g = gross cross sectional area (mm^2)

3) Maximum reinforcement ratio:
According to ACI section (10.9.1), the maximum reinforcement ratio shall be 8% of the gross cross sectional area. Thus,

$$A_{st,max} = 0.08\ A_g \tag{4.51}$$

4) Dimension compatibility between upper and lower column:
The dimensions of the upper column shall be smaller than the dimension of the column below, to maintain structural integrity and ensure the continuity of reinforcement between bottom and top columns. Thus,

$$b_{top} \leq b_{bottom} \tag{4.52}$$

And,

$$h_{top} \leq h_{bottom} \tag{4.53}$$

Where b_{top} = width of top column (mm), b_{bottom} = width of bottom column (mm), h_{top} = length of top column (mm), h_{bottom} = length of bottom column (mm).

5) Dimension compatibility between column and beams at joints:
The width of beams framing into a joint shall not exceed the width of the column in the same story. Thus,

$$b_{beam} \leq b_{column} \tag{4.54}$$

Where b_{beam} = width of beam (mm), b_{column} = width of column (mm).

6) Stiffness and slenderness of columns:
For column reinforcements to be effective in resisting bending, the lever arm should be as large as possible, thus the length of a column should be adequate as well as its moment of inertia. This can be achieved by ensuring that the in plane dimension of columns are larger than the other dimension. Thus,

$$h_{column} \geq b_{column} \tag{4.55}$$

Where h_{column} = length of column in the plane of the frame (mm).

Furthermore, an upper limit for slenderness was established to avoid the use of very slender columns, which was taken to be 100, unless stated otherwise. Thus,

$$\frac{kl_u}{r} \leq 100 \tag{4.56}$$

7) Minimum longitudinal bar spacing:

According to ACI section (7.6.3), in tied reinforced compression members, clear distance between longitudinal bars shall not be less than 1.5 the bar diameter or 40mm.

4.4.3 Beam Constraints

The following constraints were considered in the design of beams:

1) Moment Strength Constraint:

The beam shall have adequate strength to resist the applied factored bending moment. That is,

$$\phi M_n \geq M_u \tag{4.57}$$

Where $\emptyset$ = Strength reduction factor, M_n = Nominal bending moment resistance of a reinforced concrete beam (kN.m), M_u = Ultimate applied bending moment (kN.m).

2) Maximum Shear force carried by reinforcement:

According to ACI section (11.4.7.9),

$$V_{s,max} = 0.66\sqrt{f_c'}\, b_w d * 10^{-3} \tag{4.58}$$

Where $V_{s,max}$ = maximum shear force carried by shear reinforcement (kN), b_w = web thickness (mm).

3) Minimum reinforcement area:

According to ACI section (10.5.1), the minimum reinforcement shall be,

$$A_{s,min} = \frac{0.25\sqrt{f_c'}}{f_y} b_w d \geq \frac{1.4 b_w d}{f_y} \tag{4.59}$$

4) Minimum ductility level for beams:

According to ACI section (10.3.5), for non prestressed flexural members and non prestressed flexural members with factored axial compressive load less than $0.10 f_c' A_g$, the tensile strain in extreme tension steel shall be more than 0.004.

5) Minimum bar spacing:

The ACI code specifies limits for bar spacing to permit concrete to flow smoothly into space between bars without honeycombing. ACI section (7.6.1) and (3.3.2) specify the minimum clear spacing between bars as follows:

$$S_{min} = larger\ of \left(25mm, d_b, \frac{4}{3} Max.Agg.Size\right) \tag{4.60}$$

Where *Max. Agg. Size* = Maximum Aggregate Size in concrete mix (mm).

6) Ratio between cut off bars and continuous bars:

According to ACI section (12.11.1), at least one third the positive reinforcement in simple members and one fourth for positive reinforcement in continuous members shall extend along the same face of member into support. In beams, such reinforcement shall extend into the support at least 150mm.

7) Deflection control:

For deflection control purposes, the minimum height of beams can be conservatively taken from Table (4.2) (Akin & Saka, 2011).

4.4.4 Joint Constraints

The only constraint considered for joints is the anchorage length required for beam bars to anchor inside the joint core. The anchorage length is discussed in section (4.3.5) given by Eq.(4.46).

4.5 Concluding Remarks

The previously mentioned design procedure, equations and constraints were used in the development of the program that handles the optimization of reinforced concrete frames in the research works presented in this thesis. The next chapters shall discuss in detail the structure of the optimization program and the formulation of the optimization problem.

CHAPTER 5: FORMULATION OF OPTIMIZATION PROBLEM

5.1 Introduction

Formulation of an optimum design problem involves transcribing a verbal description of the problem into a well-defined mathematical statement. A set of variables to describe the design, called design variables, are given in the formulation. All designs have to satisfy a given set of constraints, which include limitations on sizes, strengths and response of the system. If a design satisfies all constraints, it is accepted to be as a feasible design. A criterion is needed to decide whether or not one design is better than another. This criterion is called the objective function. A general flowchart for any optimization procedure is shown in Figure(5.1) (Arora, 1989).

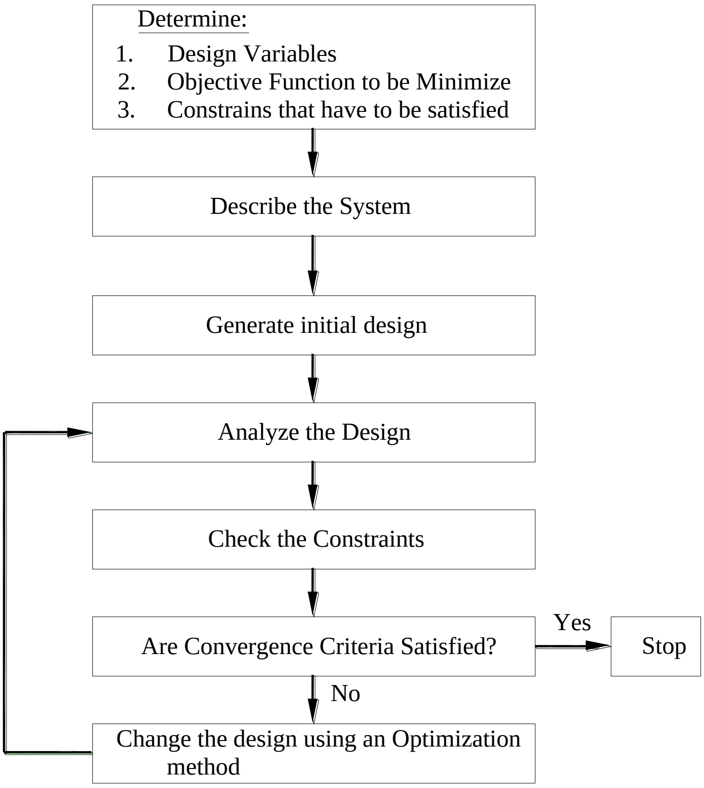


Figure 5. 1: General optimization flowchart

5.2 Design Variables

Any solution for an optimization problem is consisted of a series of variables that define a given frame in terms of section dimensions, reinforcements, number of stories and number of bays. These variables are discussed in the next sections.

5.2.1 Beam Design Variables

The design variables of a beam describe its cross section characteristics as well as its reinforcement arrangements. Some of these variables depend on the number of spans (N_{Span}), such as the number of bottom reinforcement bars, while others don't. The design variables for beams in frames under gravity loads are shown in Figure (5.2).

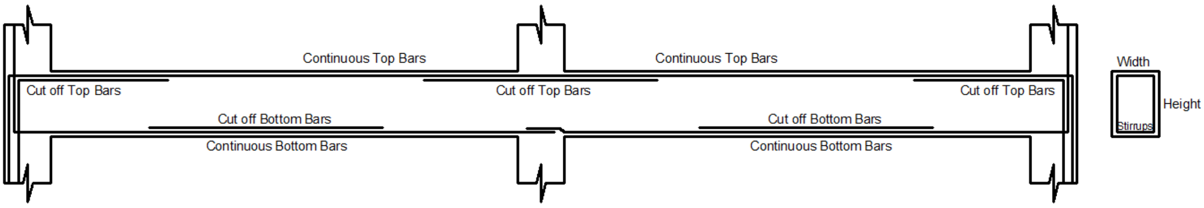


Figure 5. 2: Beam design variables

Table (5.1) further summarizes these variables:

Table 5. 1: Design variables used for beams (General)

Variable type	Unit	Total number of variables for a single beam	abbreviation
Width of beam	mm	1	b
Height of beam	mm	1	h
Number of continuous bottom reinforcement	-	N_{Span}	N_{cb}
Diameter of continuous bottom reinforcement	mm	N_{Span}	d_{cb}
Number of cutoff bottom reinforcement	-	N_{Span}	N_{pb}
Diameter of cutoff bottom reinforcement	mm	N_{Span}	d_{pb}
Number of continuous top reinforcement	-	1	N_{ct}
Diameter of continuous top reinforcement	mm	1	d_{ct}
Number of cutoff top reinforcement	-	N_{Span} +1	N_{pt}
Diameter of cutoff top reinforcement	mm	N_{Span} +1	d_{pt}
Diameter of Stirrups	mm	1	d_s*
Total number of Variables		$6N_{Span} + 7$	

* the diameter of stirrups are taken the same for both columns and beams

5.2.2 Column Design Variables

The design variables of columns, as shown in Figure (5.3), describe its cross section and reinforcement characteristics, and are used to tell the optimizer how the reinforcement bars are distributed.

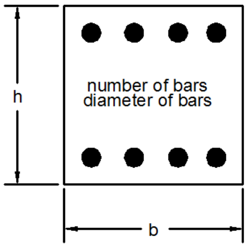


Figure 5. 3: Column design variables

Three possible bar configurations for columns were considered as shown in Figure (5.4), the selection of a certain bar configuration depends on the available spacing of bars. If there is sufficient spacing, all bars are arranged in two layers, giving the highest moment arm and thus the highest bending resistance. If the spacing is not enough, one intermediate layer is considered. Otherwise, two intermediate layers are considered.

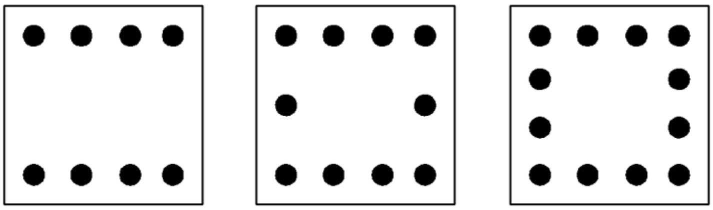


Figure 5. 4: Possible bar configurations for columns

Table (5.2) further summarizes the design variables of columns:

Table 5. 2: Design variables used for columns (General)

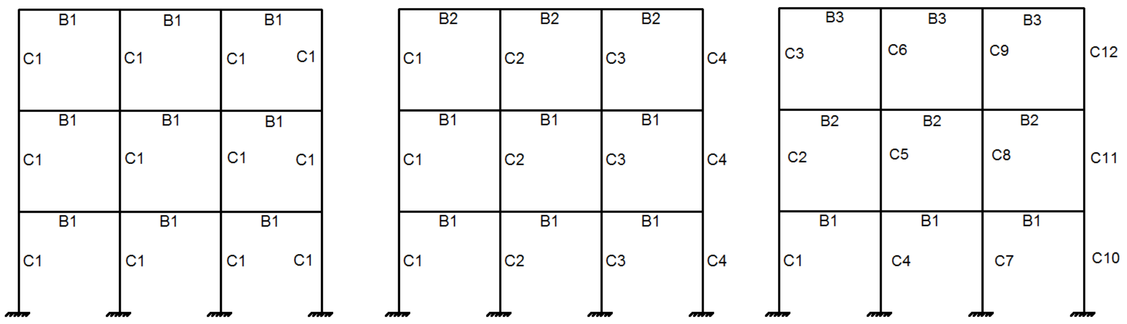
Variable type	Unit	abbreviation
Width of column	mm	b_c
Height of column	mm	h_c
Number of reinforcing bars used	-	n_b
Diameter of reinforcing bars used	mm	d_b
Diameter of Stirrups	mm	d_s*
Total number of Variables		4**

* the diameter of stirrups are taken the same for both columns and beams

** The diameter of stirrups were not considered in the total number of variables since it was already considered in the total number of variables for beams.

5.2.3 Design Groups

The use of design groups in engineering for columns and beams is well known and used to achieve a more economical design without overcomplicating the design problem. To understand the concept of grouping, consider the three frames shown in Figure(5.5).



(a) – Grouping 1 (b) – Grouping 2 (c) – Grouping 3

Figure 5. 5: The concept of design groups in reinforced concrete frames

Grouping scheme 1, shown in Figure(5.5.a), considers all columns in all stories to be the same, thus the edge and interior columns have the same design. This is inadequate since edge columns are characterized by high moment and low axial force whereas interior columns are characterized by low moment and high axial force. Such difference requires columns of different loading characteristics to be designed independently. On the other hand, grouping scheme 3 in Figure(5.5.c) shows the total opposite of the first frame, it considers every single column to be of different dimensions than the column in any other story. Although this approach obtains the most economical design, it complicates the design problem to such an extent that it the extra savings in cost are not enough for the extra effort required designing such a large number of groups. The grouping scheme 2 in Figure(5.5.b) shows a somewhat balanced case, in which the number of design groups are selected to meet both economic considerations and simplicity of the design problem. Therefore, in the research work, the concept of grouping is used. Each group has its own design variables that are independent of those in other groups. Thus, the total number of design variables for a frame can be calculated as follows:

$$D = 4N_{CG} + (6N_{Span} + 7)N_{BG} \qquad (5.1)$$

Where N_{CG} = Number of column groups, N_{BG} = Number of beam groups

5.2.4 Utilization of Symmetry

The concept of symmetry can reduce the number of possible design options significantly. A structure can be assumed symmetric if the characteristics and loading conditions on one half are identical of those on the other half. Figure(5.6.a) shows a structure in which symmetry cannot be considered whereas Figure(5.6.b) shows a structure in which symmetry can be considered.

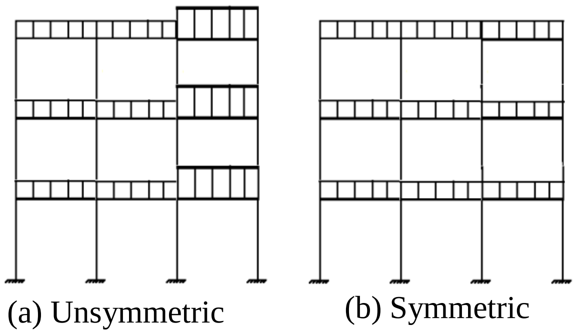


Figure 5. 6: Symmetry in Structures.

5.2.5 The Representation of a Possible Solution

Any reinforced concrete frame solution is represented using an array format, since the programming language used (Matlab) deals with highest efficiency with array and matrix formats. The array contains all information necessary to fully define the sections, reinforcement sizes and reinforcement arrangements within a frame. Figure(5.7) shows the general format of the array representation of columns, whereas Figure (5.8) shows that of one beam group.

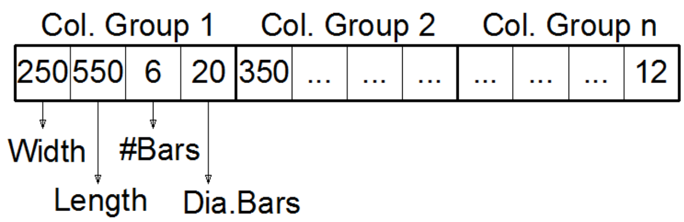


Figure 5. 7: Array representation for column design variables

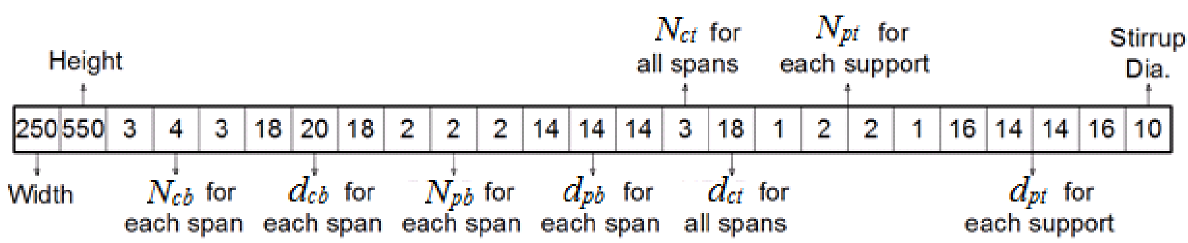


Figure 5. 8: Array representation for beam design variables for one beam group

5.3 Optimization Problem

The optimization objective is to find the best adequate frame with minimum cost. The term "adequate" implies that this frame shall have sufficient strength and deformation characteristics as well as meeting any other constraints set forth in the formulation of the optimization function. The mathematical formulation of the objective function as well as the constraints will be discussed in sections 5.4, 5.5 and 5.6.

5.4 The Objective Function

The objective of this optimization is to minimize the cost of the frame while still satisfying the strength and serviceability of the ACI318-08 code, which were stated in chapter(4). Therefore, the optimum design problem can be stated as follows:

$$Minimize\ F(x) = F_b + F_c \tag{5.2}$$

$$Where\ F_b = \sum_{i=1}^{N_b} C_c.(V_{it} - V_{is}) + C_s V_{is} \gamma_{is} + C_f.A_{fb} \tag{5.3.a}$$

$$And\ F_c = \sum_{i=1}^{N_c} C_c.(V_{it} - V_{is}) + C_s V_{is} \gamma_{is} + C_f.A_{fc} \tag{5.3.b}$$

$$Subject\ to\ C \leq 0\ where\ C = \sum_{i=1}^{n} c_i \tag{5.4}$$

Where:

- $F(x)$ = the Objective Function, which represents the total cost of the frame ($)
- F_b = total cost of beams in the frame ($)
- F_c = total cost of columns in the frame ($)
- N_b = total number of beams in a frame
- N_c = total number of columns in a frame
- C_c = cost of concrete ($/m^3)
- C_s = cost of Steel ($/kN)
- C_f = cost of formwork ($/m^2)
- V_{it} = total volume of member (m^3)
- V_{is} = volume of steel reinforcement in a member (m^3)
- A_f = total formwork area (m^2)
- γ_s= weight per unit volume of steel (kN/m^3)
- C = Penalty (constraint violation) function, which is the sum of all constraint violations
- c_i = violation function of a specific constraint
- n = total number of constraints for a given frame.

5.5 Penalized Objective Function

In order to assess the fitness of a trial design and determine its distance from the global optimum, the eventual constraint violation should be computed by means of a penalty function. The penalty function consists of a series of geometric constraints corresponding to the dimensions and shape of the cross sections, and a series of constraint related to the deflection and internal forces of the members of the structure. Thus, penalty will be proportional to constraint violations, and the best design will have the minimum cost and no penalty.

The penalized objective function measures how good a solution is and can be expressed as follows,

$$\varphi(x) = \mathrm{F(x)}.[\,1 + KC]^{\varepsilon} \tag{5.5}$$

Where $\varphi(x)$= Penalized Objective Function ($), K = Penalty function constant, ϵ = Penalty function exponent. In this study, K = 1.0, ϵ = 2.0 which was also recommended by (Kaveh & Sabzi, 2011).

5.6 Penalty Function

The constraint violation is as follows:

$$C = \sum_{i=1}^{n} c_i \tag{5.6}$$

Where c_i = the violation function of a specific constraint.

In this study, a total of 20 constraints have been set for a frame to be considered adequate. The theoretical background of these constraints have already been discussed in section(4.4). This section discusses the calculation of the penalty function from these constraints.

5.6.1 Overall Frame Constraints

The only constraint put on the overall performance of the reinforced concrete frame being optimized is the maximum lateral inter story drift allowed, and is calculated as follows:

$$c_1 = \frac{\Delta_{rel} - \Delta_{rel.max}}{\Delta_{rel.max}} \geq 0 \tag{5.7}$$

Where c_1 = interstory drift penalty, $\Delta_{rel.max}$ = maximum permissible interstory drift (m) (0.020 * Story Height), Δ_{rel} = actual interstory drift (m)

5.6.2 Column Constraints

The constraints for column design include strength constraints, size constraints, reinforcement constraints as well as slenderness constraints. These constraints are presented as follows:

1) Axial Strength: The column's axial strength $\emptyset P_n$ shall be larger than the applied factor load P_u, thus the axial strength constraint c_2 is calculated as follows:

$$c_2 = \frac{P_u - \emptyset P_n}{\emptyset P_n} \geq 0 \tag{5.8}$$

2) Moment Strength: a column should have sufficient bending strength $\emptyset M_n$ to resist the applied bending moment M_u, therefore the flexural strength constraint c_3 can be calculated as follows:

$$c_3 = \frac{M_u - \emptyset M_n}{\emptyset M_n} \geq 0 \tag{5.9}$$

3) Shear Strength: a column should have sufficient shear strength $\emptyset V_n$ to resist the applied shear force V_u, therefore the shear strength constraint c_4 can be calculated as follows:

$$c_4 = \frac{V_u - \emptyset V_n}{\emptyset V_n} \geq 0 \tag{5.10}$$

4) Minimum Reinforcement Ratio: The reinforcement ratio ρ cannot be taken lower than 1%, thus the minimum reinforcement constraint c_5 is given by:

$$c_5 = \frac{0.01 - \rho}{0.01} \geq 0 \tag{5.11}$$

5) Maximum Reinforcement Ratio: The reinforcement ratio ρ cannot exceed 8% thus c_6 can be computed using:

$$c_6 = \frac{\rho - 0.08}{0.08} \geq 0 \tag{5.12}$$

6) Dimension Compatibility between top and bottom columns: its considered good engineering practice for column dimensions in top stories to be equal or lower than those in lower stories thus, dimension incompatibility constraints, namely c_7 and c_8 are calculated as follows:

$$c_7 = \frac{b_{top} - b_{bottom}}{b_{bottom}} \geq 0 \tag{5.13}$$

$$c_8 = \frac{h_{top} - h_{bottom}}{h_{bottom}} \geq 0 \tag{5.14}$$

7) Dimension Compatibility between beams and columns at joints: to avoid reinforcement and design complexities in joint regions, the width of beams are limited to those of their column counterparts, thus the width incompatibility constraint c_9 is given by:

$$c_9 = \frac{b_{beam} - b_{column}}{b_{column}} \geq 0 \tag{5.15}$$

8) Stiffness and slenderness of columns: it is recommended to select column dimensions with sufficient stiffness in the plane of bending for better reinforcement utilization and improved buckling behavior, thus two constraints, c_{10} and c_{11} are computed as follows:

$$c_{10} = \frac{b_{column} - h_{column}}{h_{column}} \geq 0 \tag{5.16}$$

$$c_{11} = \frac{\frac{kl_u}{r} - 100}{100} \geq 0 \tag{5.17}$$

9) Minimum bar spacing: spacing between reinforcement bars should be larger than a certain minimum to allow concrete to flow smoothly and avoid segregation, thus the minimum spacing constraint c_{12} is calculated by:

$$c_{12} = \frac{s_{min} - s}{s_{min}} \geq 0 \tag{5.18}$$

5.6.3 Beam Constraints

Beam constraints deal with moment capacity, adequate shear strength, reinforcement limitations as well as spacing limitations. These constraints are listed below:

1) Moment Strength: for a beam to be adequate, it must have adequate flexural strength $\emptyset M_n$ that is capable of resisting the applied moments M_u. If not, a constraint c_{13} is given:

$$c_{13} = \frac{M_u - \emptyset M_n}{\emptyset M_n} \geq 0 \tag{5.19}$$

2) Maximum Shear force on reinforcements: in no case shall the shear force carried by shear reinforcement exceed the corresponding maximum stipulated in the ACI code. If such maximum shear is exceeded, the constraint parameter c_{14} calculates such violation:

$$c_{14} = \frac{V_s - V_{s,max}}{V_{s,max}} \geq 0 \tag{5.20}$$

3) Minimum Reinforcement Area: the area of reinforcement A_{st} shall be larger than the minimum reinforcement given by the ACI code, thus,

$$c_{15} = \frac{A_{st,min} - A_{st}}{A_{st,min}} \geq 0 \tag{5.21}$$

4) Minimum ductility for reinforcement in beams: for flexural members to fail in a ductile manner, the strain in the extreme steel layer ε_t shall exceed 0.004, therefore,

$$c_{16} = \frac{0.004 - \varepsilon_t}{0.004} \geq 0 \tag{5.22}$$

5) Minimum bar spacing: adequate bar spacing in beams must be provided to allow concrete to flow smoothly and avoid segregation, thus the minimum spacing constraint c_{17} is calculated by:

$$c_{17} = \frac{s_{min} - s}{s_{min}} \geq 0 \tag{5.23}$$

6) Ratio between cut off bars and continuous bars: the number of continuous bars n_f shall be at least 0.25 of the total number of reinforcing bars n_t,

$$c_{18} = \frac{0.25 - \frac{n_f}{n_t}}{0.25} \geq 0 \tag{5.24}$$

7) Deflection characteristics: the height of a reinforced concrete beam section is one method to limit deflections in reinforced concrete beams not part of a moment

resisting frame and can be conservatively used to limit deflections in frames (Akin & Saka, 2011), thus,

$$c_{19} = \frac{h_{min} - h}{h_{min}} \geq 0 \tag{5.25}$$

5.6.4 Joint Constraints

Anchorage of reinforcement: the available length for reinforcement anchorage $l_{available}$ shall be larger than the minimum required anchorage length $l_{dh,min}$ to prevent reinforcement slippage, thus,

$$c_{20} = \frac{l_{dh,min} - l_{available}}{l_{available}} \geq 0 \tag{5.26}$$

5.7 Development of the Optimization Algorithm

The optimization algorithm required to carry out the optimization process of reinforced concrete frames is described in the following sections. It describes the programming language used, fixed parameters as well as the algorithm's flowchart. The programming code used can be found in Appendix (A).

5.7.1 Matlab as a Programming Language

The Matlab is a high-performance language for technical computing that integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Matlab also has a large number of additional application toolboxes for mathematical operations and analysis, data acquisitions, signal processing, control design, finance and economics and simulation.

5.7.2 Fixed Parameters of the Optimization Model

The fixed parameters used as main input in the optimization model can be divided into 5 categories, namely: structural parameters, loading and analysis parameters, material parameters, design variable limits and step sizes as well as the Artificial Bee Colony (ABC) main control parameters.

The structural parameters describe the overall shape of members within a frame such as: the number of bays, number of stories, bay length and story height. It also describes the organization of frame members, namely: the member groups used, symmetry utilization as well as the maximum permissible slenderness coefficient used for columns (stability issues).

On the other hand, the analysis parameters describe the dead load, live load, lateral load type and value, whether selfweight is taken into account or not as well as the support type (either fixed or pinned).

Other parameters initialize material properties and costs, such as: concrete compressive strength, concrete cost, yield strength of reinforcement, reinforcement cost and formwork cost. The cost parameters also include options to control what cost elements are included or excluded from the overall frame cost, such as: shear reinforcement cost, formwork costs as well as extra costs due to joint detailing.

The optimization model user must describe the design variable limits and step sizes. Each design variable has its corresponding minimum and maximum value, as well as its own step size. One has to initialize these fixed parameters so that the design variables are kept within required limits and are selected according to a logical step size.

Finally, the control parameters of the ABC algorithm are initialized, these control parameters were discussed in section (3.5) and are: Number of bees (N_p), The improvement limit for a solution (I_L), maximum number of iterations (I_{max}), variable changing percentage (*VCP*) as well as the number of independent runs (*r*).

5.7.3 The Main ABC Algorithm Flowchart

The main ABC algorithm Matlab function file has 4 stages, namely: initialization stage, employed bee stage, onlooker bee stage and scout bee stage. The description of each stage is presented as follows:

1) The Initialization Stage:
    a. All fixed parameters are read into the model.
    b. The initial group of solutions is generated. The number of solutions generated (N_S) is equal to half of the total bees used in the problem, that is: ½ N_P.
    c. The randomly generated solutions are made symmetric if the problem is considered symmetric.
    d. The Penalized cost is calculated, and the best solution of all randomly generated solutions is stored.
2) Employed Bee Stage:
    a. Each of the N_S original solutions is used to derive a mutant solution by changing a particular number of variables, namely (*VCP*).
    b. The generated mutant solution is checked and modified to conform with the upper limit, lower limit and step size that are set.
    c. The mutant solution is made symmetric if required.
    d. The Penalized cost for the mutant solution is calculated and compared with the original solution's penalized cost. If the mutant was found better than the original, the original is replaced by the mutant and its trial counter is set back to 0. If not, the original solution is not replaced and its trial counter is incremented by 1. The trial counter represents the number of trials the algorithm performed on a certain solution to derive and improved version of the original solution. The counter will be used to determine whether a solution is considered abandoned or not.

e. The Best Solution is updated if any solution was found better than the current best.

3) Onlooker Bee Stage:
   a. The probability of each of the solutions to be selected by an onlooker bee is calculated. The higher the fitness of a solution the higher the probability this solution is selected as a target by an onlooker bee.
   b. Onlooker bees are dispatched to the solutions taking into consideration the probability of each solution, and the same steps of the employed bee phase are done the solution being chosen by an onlooker bee, that is: new solution is derived and compared to the original solution. If the new solution is better than its original, the original is replaced by the new solution. Otherwise, the trial counter is incremented.
   c. The process of dispatching onlooker bees to their destination solutions is continued until all of the onlooker bees have been dispatched.
4) Scout Bee Stage:
   a. The model checks if any solution has exceeded its trial limit in order to declare the solution as abandoned.
   b. If a solution is declared abandoned, it means that there is no longer any hope in it to improve and this solution is deleted and replaced by a new randomly generated solution, which is subject to symmetry if necessary
   c. The penalized cost of the new random generated solution is calculated and stored.

The process in stage 2, 3 and 4 are considered as one iteration for the ABC algorithm. These steps are repeated until the maximum number of iterations has been reached. Figure (5.9), shows the flowchart of the main ABC algorithm used in the research work of this thesis. This optimization algorithm is considered the leading optimizer which guides and controls all other sub algorithms.

5.7.4 Optimization Program User Manual

A simplified user manual for the optimization program is discussed in Appendix (B). The user manual contains pictures and guidelines for the use of the optimization software functions and capabilities.

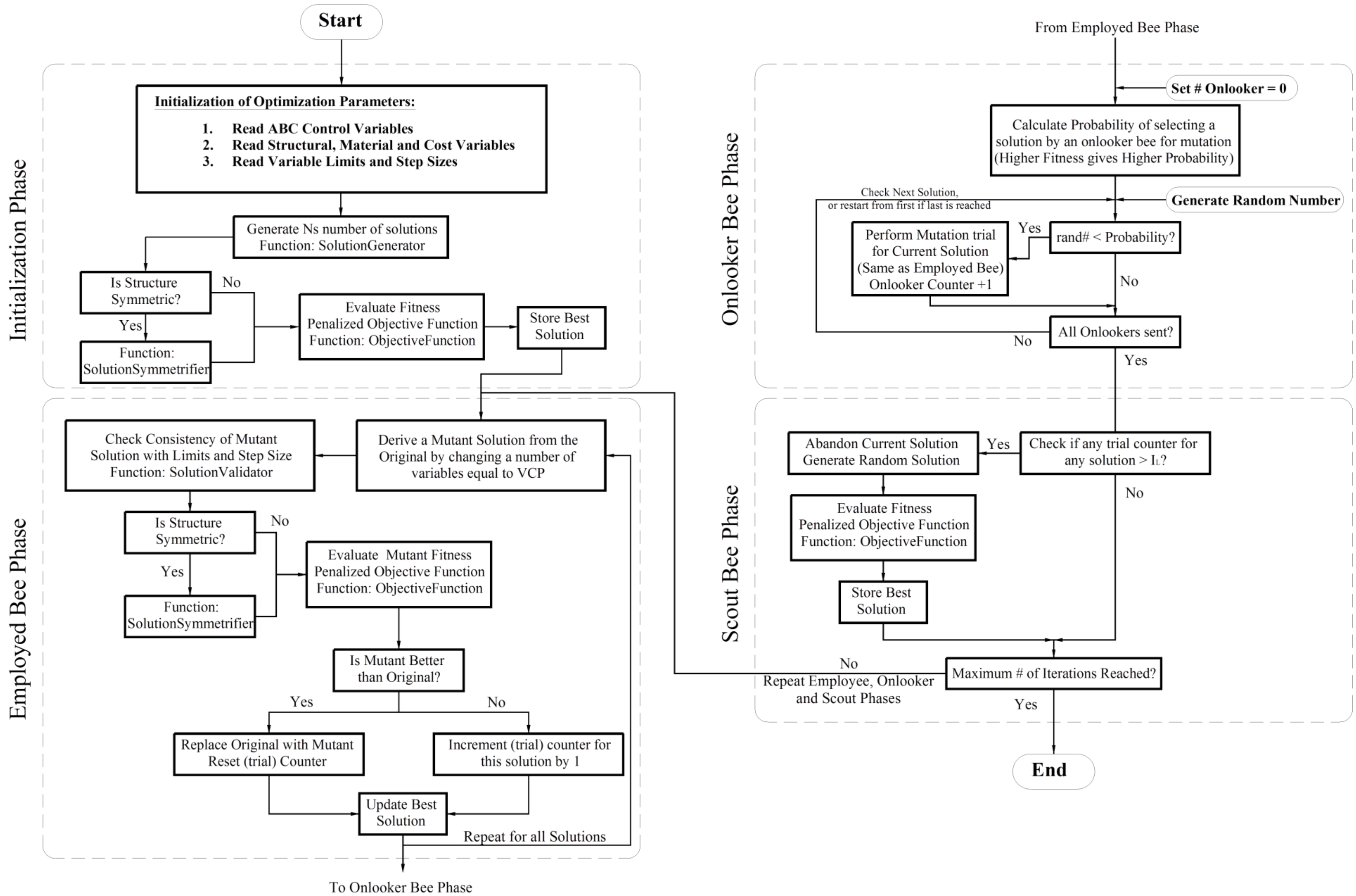


Figure 5. 9: Flowchart for the Artificial Bee Colony Algorithm

CHAPTER 6: ANALYSIS, RESULTS AND DISCUSSIONS

6.1 Introduction

The Artificial Bee Colony (ABC) Algorithm developed in the current research work is used to obtain the optimum designs of reinforced concrete frames by manipulating a pool of possible designs that result from the design variable limits. The constraints taken into account while developing the main optimization criteria are: Strength Constraints of the ACI318-08 specifications, serviceability and displacement constraints, size constraints and spacing constraints.

This chapter starts with a one bay one story frame, which acts as a test frame, to understand the influence of the various control parameters in the ABC algorithm and to find out the best parameter combination that ensures accurate optimization results with minimum deviation.

The test frame is followed by two multi bay multi story frames that were previously investigated by Kaveh and Sabzi (2011), who used two optimization techniques: The heuristic big bang-big crunch (HBB-BC), which is based on big bang-big crunch (BB-BC) and a harmony search (HS) scheme to deal with the variable constraint, and The (HPSACO) algorithm, which is a combination of particle swarm with passive congregation (PSOPC), ant colony optimization (ACO), and harmony search scheme (HS) algorithms.

The design variables used by Kaveh and Sabzi were much simpler than those used in the current study and were limited to the section dimensions as well as the number of reinforcing bars and their diameter without using cut-off bars. Such a simplification significantly decreases the design pool making optimization easier. However, such a simplification increases the total cost significantly. Furthermore, reinforced concrete frames are usually designed utilizing cut-off bars.

6.2 Optimization of One Bay One Story Reinforced Concrete Frame

A one bay one story reinforced concrete framed structure, shown in Figure (6.1), is presented here to understand the influence of the ABC algorithm's control parameter. Several ABC control parameter combination were tried and results were recorded and compared. This will result in fine tuning these parameters to obtain best results. The frame studied here is the simplest reinforced concrete frame that can be designed.

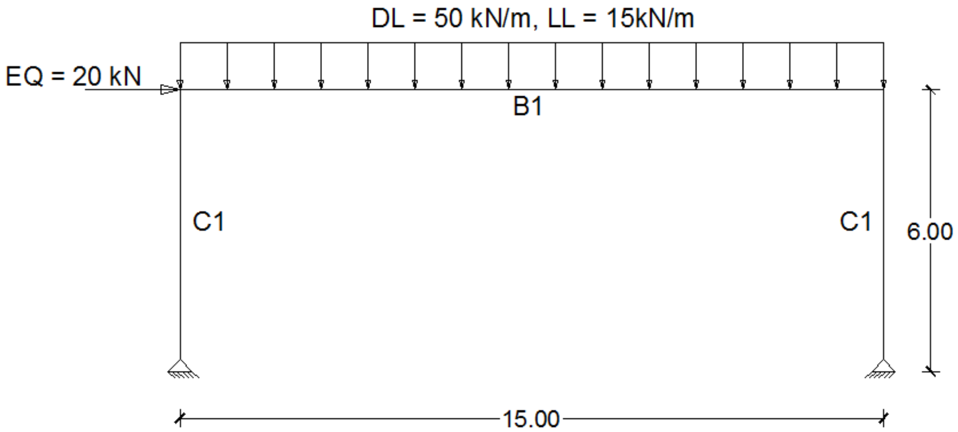


Figure 6. 1: One bay one story frame

6.2.1 Fixed Parameters of the Optimization Model

The structural parameters of the frame shown in Figure (6.1) were set as follows: the number of bays as well as number of stories was set to 1. The bay length, or span, was 15 meters whereas the story height was 6 meters. The structure was pin supported and considered symmetric. The number of stories per beam group, as well as number of stories per column group, was set to 1, which result in a grouping scheme identical to that shown in Figure (6.1).

For gravity loads, the dead load was considered 50 kN/m whereas the live load was 15 kN/m. For lateral loads, an earthquake load of 20 kN was applied on the structure. The analysis algorithm calculates member selfweights and adds it to the current dead load. Finally, the maximum permissible slenderness ratio for columns was set to 50.

For material and cost parameters, the following inputs were used: Concrete compressive strength was set to 28 MPa and costs 120 $\$/m^3$ whereas the yield strength of reinforcement was 420 MPa and costs 87 $/kN. Other cost components were included in calculating the total cost of the frame, namely: the cost of shear reinforcement, extra steel for joint detailing as well as the formwork cost. The formwork cost per unit area was set to 10 $\$/m^2$. The unit weight of concrete was 25 kN/m^3 and for steel it was 78 kN/m^3.

The design variable limits and corresponding step sizes were given to the algorithm. These limits are required to keep the design variable values within a suitable range. Once those limits as well as their corresponding step sizes are set, one can calculate the number of possible values a design parameter can use, which is used to calculate the number of possible frame designs, also called the design space, for the design problem at hand. Table (6.1) presents the variable limits and step sizes for beams, whereas Table (6.2) shows those used for columns.

Table 6. 1: Design variables for beams (One Bay One Story)

Variable Name	Unit	Lower Limit	Step Size	Upper Limit	# Possible Variables Values
Beam Width	mm	300	50	700	9
Beam Height	mm	500	50	1500	21
Number of continuous bottom reinforcement	-	2	1	10	9
Diameter of continuous bottom reinforcement	mm	12	2	22	6
Number of cutoff bottom reinforcement	-	0	1	10	11
Diameter of cutoff bottom reinforcement	mm	12	2	22	6
Number of continuous top reinforcement	-	2	1	10	9
Diameter of continuous top reinforcement	mm	12	2	22	6
Number of cutoff top reinforcement	-	0	1	10	11
Diameter of cutoff top reinforcement	mm	12	2	22	6
Diameter of Stirrups	mm	10	2	12	2

Table 6. 2: Design variables for columns (One Bay One Story)

Variable Name	Unit	Lower Limit	Step Size	Upper Limit	# Possible Variables Values
Column Width	mm	300	50	700	9
Column Depth	mm	250	50	1000	16
Number of Reinforcing Bars used	-	4	2	20	9
Diameter of Reinforcing Bars used	mm	12	2	22	6

To finalize, the ABC optimization algorithm requires its main control parameters to be set. But as mentioned in the beginning of this section, several control parameter combinations were tried in order to fine tune these parameters for further optimization problems.

6.2.2 The Design Space

Although the frame considered here is rather simple, the number of possible frame solutions, however, is practically considered infinite for the given possible variable combinations. The number of possible frame solutions in the design pool is calculated below for both symmetric and non-symmetric structure. Although the frame is considered symmetric, the design pool for a non-symmetric frame is calculated for comparison purposes.

For Symmetric Structures:

1) Columns:
   a. Number of Possible Columns per Group = 9 x 16 x 9 x 6 x 2 = 15552 Possible Design.
   b. Number of Column Groups = 1
   c. Total Number of Possible Column Designs = 15552^1 = 15552 Possible Design
2) Beams:
   a. Number of Possible Beam Designs per Group =
      9 x 21 x 9 x 6 x 11 x 6 x 9 x 6 x 11 x 6 x 2 = 4801392288 Possible Design
   b. Number of Beam Groups = 1
   c. Total Number of Possible Beam Designs = 480139228^1 = 480139228 Possible Design
3) Total Possible Frame Designs = 480139228 x 15552 = 7.46 x 10^{13} Possible Frame Design

For Non-Symmetric Structures:

1) Columns:
   a. Number of Possible Columns per Group = 15552 Possible Design.
   b. Number of Column Groups = 2
   c. Total Number of Possible Column Designs = 15552^2 = 241864704 Possible Design
2) Beams:
   a. Number of Possible Beam Designs per Group = 4801392288 Possible Design
   b. Number of Beam Groups = 1
   c. Total Number of Possible Beam Designs = 480139228^1 = 480139228 Possible Design
3) Total Possible Frame Designs = 480139228 x 2418674704 = 1.16 x 10^{18} Possible Frame Design

It is clear that symmetry should be taken into consideration whenever possible, since it decreases the number of possible frame designs from 1.16 x 10^{18} to 7.46 x 10^{13}.

Of course, the number of possible frame designs even after taking symmetry into consideration is still huge and may be considered close to infinity. These numbers are much higher when considering multi bay and multi story frames such as the frames presented later on in the research works of this thesis.

6.2.3 The ABC Algorithm Control Parameters

The purpose of this test frame, as mentioned before, is to obtain a set of control parameters that should work best for this test frame and could be applied, with little or no modification, on other frames discussed in the research work of this thesis. The influence of each of the control parameters is discussed in the following paragraphs, followed by the range in which these parameters were tested.

6.2.3.1 The Number of Bees in the Colony (N_P)

The number of bees in a colony of artificial bees corresponds to the number of solutions being investigated each iteration. A large number of bees would mean that a large number of frames are simultaneously investigated but also means a higher computational effort is required. The number of bees should be related to the complexity of the problem, that is, the more complex the problem the higher the number of bees working on it.

Several tests were carried out on different number of bees. The number of bees taken were: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 and 110. The results of various values of N_P are discussed in section (6.2.4.2)

6.2.3.2 The improvement limit for a solution (I_L)

As previously discussed in chapter (3), each solution undergoes some modifications to derive a new solution that is somehow different than the parent solution. Sometimes, however, parent solutions are placed in bad regions of the design space making further improvement on such solutions hard to achieve. Thus a predetermined number of trials to improve a solution, or improvement limit (I_L) should be set to avoid being stuck at local minima. Once the predetermined number of trials is achieved, the solution is deleted and replaced by a totally different one. The value of I_L was investigated by (Karaboga & Basturk, 2008), who recommended the value of ($N_P/2 * D$) to be adequate, where D is the dimension of solution vector which corresponds to the number of design variables used to define a certain structure.

6.2.3.3 Maximum Number of Iterations (I_{max})

The maximum number of iterations is used to set a stopping criterion for the ABC algorithm. The maximum number of iterations shall be adequate to give the ABC algorithm sufficient time to converge. This parameter was taken as 2000.

6.2.3.4 Variable Changing Percentage (*VCP*)

This is another important parameter in the ABC algorithm. In the original algorithm, no such parameter was present since each member was defined by a single variable (such as standard steel sections). In reinforced concrete, however, a member is defined by a series of variables that define its geometry, reinforcement size and reinforcement arrangement. Thus, more than one variable needs to be changed in the solution to derive a mutant that is somehow different than the original. Several tests were run on different values of (*VCP*). These parameters were taken to be: 10%, 20%, 30%, 40%, 50%, 60% and 70% of the total number of variables in a solution vector. The results for various values of (*VCP*) are discussed in section (6.2.4.1).

6.2.3.5 Number of independent runs (*r*)

Ten independent runs were performed for each of the control parameter configuration in order to obtain representative results and analyze the deviation of these results.

6.2.4 Optimization Results and Discussion

The results of various control parameter combinations are presented in this section. The test runs were carried out using different values for two control parameters, namely: The number of bees in a colony (N_p) and the Variable changing percentage (*VCP*).

6.2.4.1 Effect of Variable Changing Percentage (*VCP*) on Optimization Results

The optimization process was conducted using several values for *VCP*, namely, 10%, 20%, 30%, 40%, 50%, 60% and 70%. The other ABC control parameters were set constant: The number of bees (N_P) was 70, the improvement limit (I_L) was 735 and the maximum number of iterations (I_{max}) was 2000.

For each value of *VCP*, 10 independent runs were conducted, which means that a total of 70 test runs were conducted. For each of the 70 test runs, the optimum cost as well as the iteration at which this cost was obtained, called last improvement, is recorded. The value of last improvement measures how fast the algorithm converges to the optimum solution. The results for the 70 test runs are listed in Appendix (C).

Table (6.3) summarizes the results of the 70 independent test runs carried out. As mentioned before, for each *VCP* a total number of 10 independent runs was carried out. The lowest optimum cost, called Best Cost, of all 10 independent runs is listed. Furthermore, The average of the optimum costs, called Average Cost, obtained by the 10 independent runs is also given. Table (6.3) also gives the standard deviation of the optimum cost of each of the 10 independent from their corresponding average. Finally, the average value of the last improvement is also calculated and given. Table (6.3) also illustrates the effect of changing the parameter *VCP* on the average optimum cost retrieved from the 10 test runs. The results for using a value of 10% of *VCP* indicate poor performance, which is evident by the high Average Cost, Standard Deviation and Average Last Improvement. This lack in performance is due to the slow convergence rate of the algorithm using low values for *VCP*, since members – as discussed in section (5.2.5) – are represented by a series of variables, which means that for a new solution to be significantly different than its parent, sufficient variables need to be changed. Also, the results for high values of *VCP*, namely 60% and 70%, indicate even worse performance especially for Standard Deviation and Average Cost, since high values of *VCP* tend to totally randomize the search without utilizing the good properties of parent solutions sufficiently.

Table 6. 3: Summary of test programs for different values of *VCP*

Test Program	*VCP*	Average Last Improvement	Average Cost ($)	Best Cost ($)	Standard Deviation ($)
1	10%	1621	3546.88	3536.5	26.30
2	20%	1355.4	3536.81	3536.5	0.98
3	30%	1238.1	3536.81	3536.5	0.98
4	40%	1033.5	3536.81	3536.5	0.98
5	50%	839.3	3538.36	3536.5	1.60
6	60%	694.3	3643.74	3536.5	163.56
7	70%	523.6	3701.64	3536.5	176.51

It can be seen that the minimum average cost of all seven test programs is 3536.81$, with a best cost for a single run of 3536.5$. The variation between the Average Cost of the test programs versus the values of *VCP* is illustrated in Figure(6.2). Test program 2, 3 and 4 indicated relatively good performances. All these three test programs had the same standard deviation and average cost. However, the Average Last Improvement of the three test programs varies, with a minimum average of 1033.5 iterations corresponding to a value of 40% for *VCP*. The standard deviation of 0.98$ is considered very good for a design problem of such a large magnitude (7.46 x 10^{13} possible frame designs) and reflects accuracy, precision and robustness of the ABC search algorithm in dealing with large design spaces.

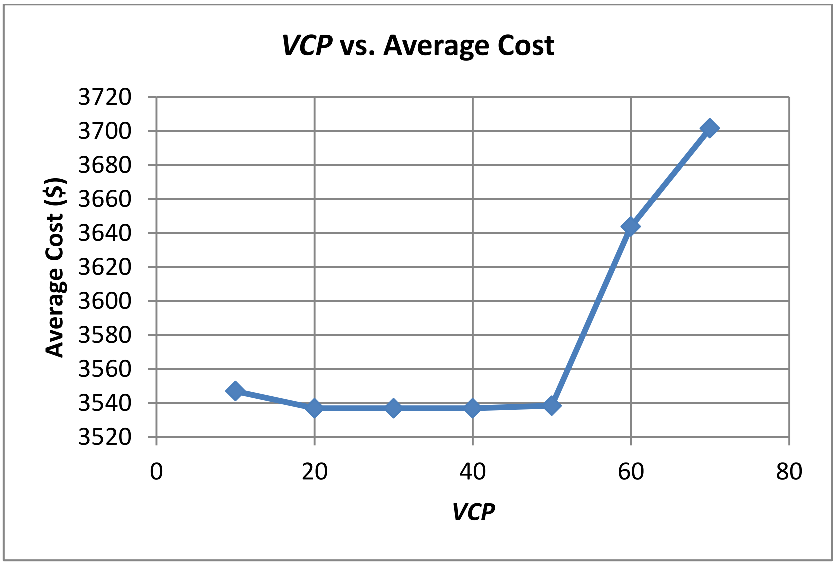


Figure 6. 2: Variation of Average Cost with different values of *VCP*

The convergence history of the best test program, which uses a value of 40% of *VCP*, is shown in Figure(6.3). The figure shows a typical convergence history that starts with a very steep slope then goes into a somehow steady state characterized by its low slope, since the program arrives to near optimal costs making further improvement harder. Figure (6.4) focuses on the first 200 iterations to further illustrate the transition from steep slope to steady state.

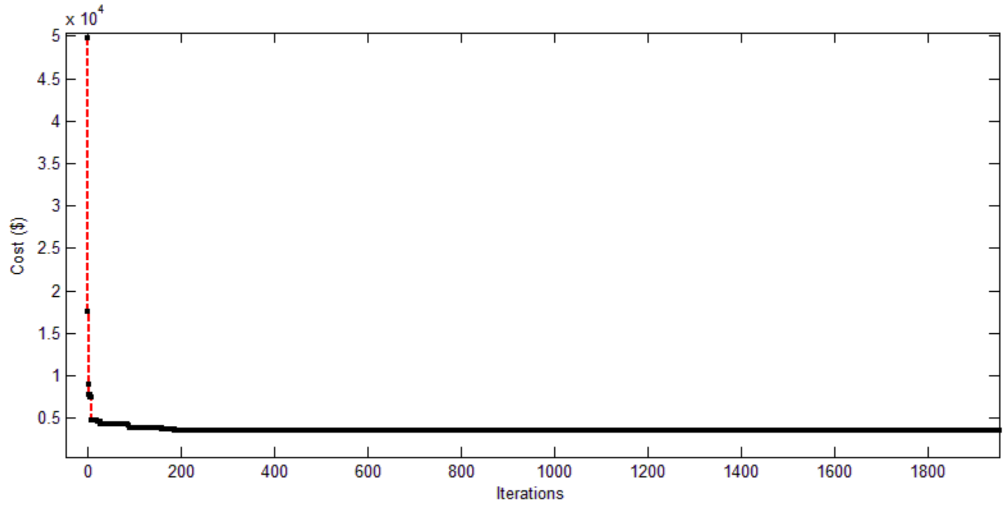


Figure 6. 3: Convergence history of optimization at 40% *VCP*

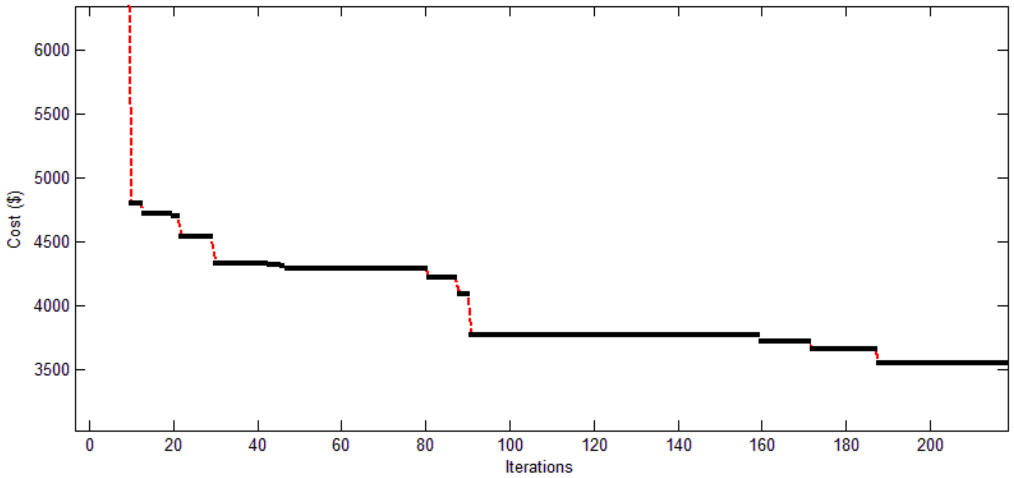


Figure 6. 4: Transition from steep slope to steady state at 40% *VCP*

The best cost at the 100th iteration was 3773.3$, which is 6.7% larger than the optimum cost of 3536.5$. However, at the 200th iteration, which is considered 10% of the total number of iterations, the cost was 3552.6$, which is 0.4% larger than the optimum cost of 3536.5$, which concludes that the ABC algorithm is a fast converging high performance algorithm that is capable of reaching near optimal solution 0.4% difference at iterations as low as 10% of the total number of iterations.

From the optimization of the test frame using different values for the control parameter *VCP*, one can conclude that the optimum value for this control parameter is 40% of the size of array representation of the test frame. This percentage is taken the same for all frames in the current research works.

6.2.4.2 Effect of Number of Bees (N_P) on Optimization Results

Different values for the parameter N_P, discussed in section (6.2.3.1), namely 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 and 110, were tried. Other ABC control parameters were set constant as follows: The variable changing percentage (*VCP*) was set to 40%, the improvement limit (I_L) was 735 and the maximum number of iterations (I_{max}) was 2000.

For each value of N_P, 10 independent runs were carried out, giving a total number of 110 test runs. For each of the 110 test runs, the optimum cost and corresponding iterations is recorded. This section shows the summary of test results. For detailed test results of the 110 test runs, one can refer to Appendix (C).

The control parameter N_P is directly proportional to the calculation effort required for each iteration of the ABC algorithm; since higher the value of N_P correspond to higher number of frames being simultaneously investigated. The number of frames investigated at the point the optimum solution is obtained, called Total Frames Tried, is yet another influencing factor in determining which value of N_P should be chosen as a compromise between efficiency and computational effort (or time). The results for each test program are summarized in Table (6.4).

Table 6. 4: Summary of test programs for different values of N_P

Test Program	N_P	Average Last Improvement	Average Cost ($)	Best Cost ($)	Total Frames Tried	Standard Deviation ($)
1	10	201.1	4503.0	3856.6	2011	721.6
2	20	610.5	3935.6	3579.2	12210	308.8
3	30	844.7	3555.3	3536.5	25341	24.5
4	40	910.3	3601.1	3536.5	36412	103.2
5	50	856.8	3583.9	3536.5	42840	98.0
6	60	974.9	3542.6	3536.5	58494	18.4
7	70	1033.5	3536.8	3536.5	72345	0.98
8	80	883.7	3537.3	3536.5	70696	1.27
9	90	822.3	3536.9	3536.5	74007	1.01
10	100	754.6	3536.5	3536.5	75460	0
11	110	714.6	3536.5	3536.5	78606	0

Table (6.4) illustrates the effect of N_P on the average optimum cost retrieved from the 10 test runs. The results obtaining using very small values of N_P indicate poor performance, which is evident by the high Average Cost, Standard Deviation and Average Last Improvement. The low number of N_P decreases the diversity of the sample being optimized and decreases the probability of obtaining better solutions each iteration, thus increasing the value of N_P should yield better results. Increasing N_P increases the diversity of the solutions being optimized, and thus, yields better results. The side effect of such an increase in diversity, however, is the fact that a larger number of frames are simultaneously investigated in each iteration. Thus the computational effort in dealing with such an increased number of frames grows, making optimization slower. It is simply a tradeoff between effort and cost savings.

For test programs 7 to 11, it can be noted that the standard deviation decreases with the increase of the number of bees (N_P) but also increases the number of frames tried (or time). Another important conclusion for the one bay one story frame is that any increase in N_P beyond 100 has adverse effects on optimization: it does not improve the standard deviation but increases the number of frames tried, thus increasing time and effort to obtain the optimum solution.

The convergence history of test program 10, which uses a value of 100 for N_P, is shown in Figure(6.5). The convergence history starts with very high costs since, at the beginning of the test run, the ABC algorithm has no guidance and relies on complete random solutions. These random solutions don't usually satisfy the constraints set on the frame design and have large values of penalty functions, thus increasing their penalized cost. As the algorithm continues, the solutions start to converge to the optimum solutions and the penalty decreases rapidly, which can be seen by the steep slope at the beginning of the run. Later, the algorithm reaches near optimal solutions, which makes the improvement of these harder, decreasing the slope of the curve to a somehow steady state condition.

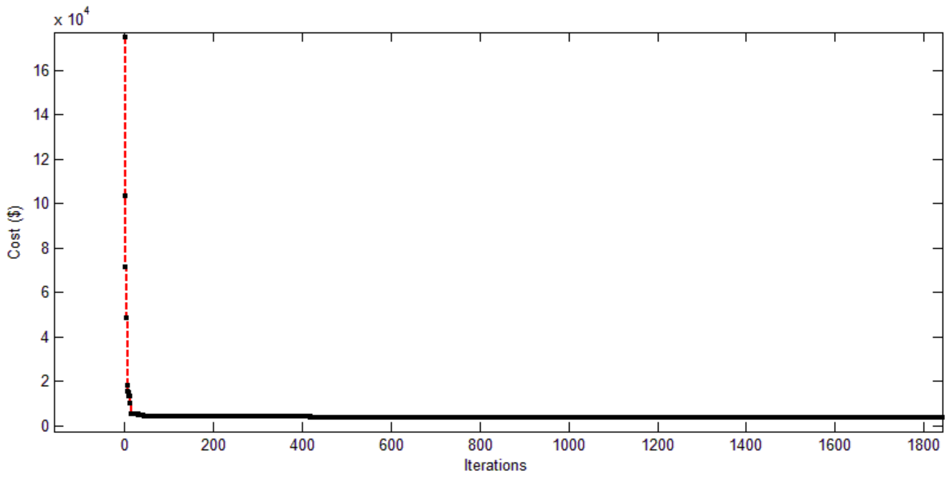


Figure 6. 5: Convergence history of optimization at N_P = 100

Figure (6.6) focuses on the first 200 iterations to further illustrate the transition from steep slope to steady state.

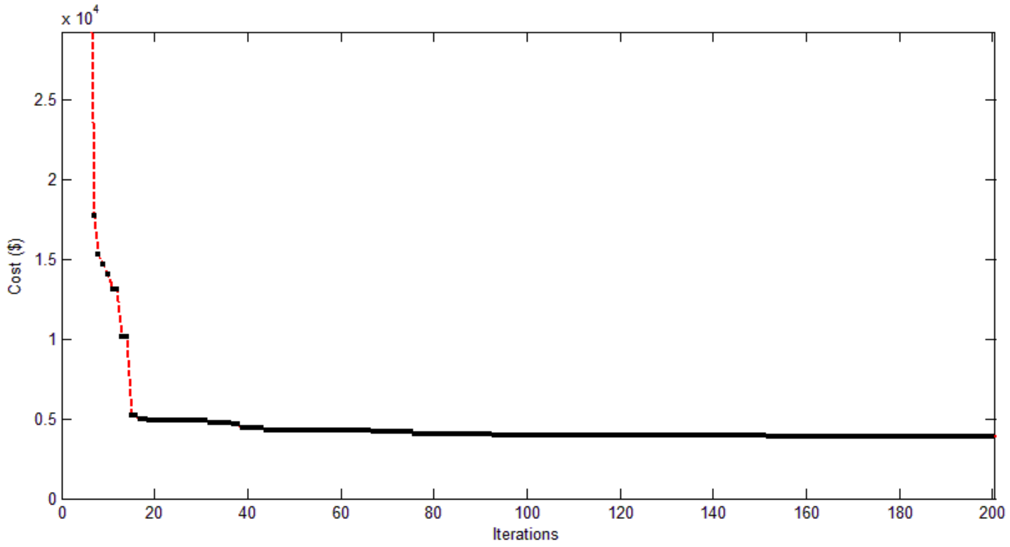


Figure 6. 6: Transition from steep slope to steady state at N_P = 100

The best cost at the 100th iteration was 3629.2$, which is 2.6% larger than the optimum cost of 3536.5$. However, at the 200th iteration, which is considered 10% of the total number of iterations, the cost was 3583.3$, which is 1.3% larger than the optimum cost of 3536.5$, which concludes that the ABC algorithm is a fast converging high performance algorithm that is capable of reaching near optimal solution 1.3% difference at iterations as low as 10% of the total number of iterations.

From the optimization of the test frame using different values for the control parameter N_P, one can conclude that:

1) If the computational effort is more important than the overall cost and deviation of the frame being optimized, one can use a lower value for N_P. In other words, the accuracy is sacrificed to obtain near optimum results for the sake of effort required to obtain such result.
2) If the overall cost is of prime priority, one can use higher values for N_P. This sacrifices computational effort for the sake of cost.

6.2.4.3 Optimum Solution Details and Characteristics

The previous two sections (6.2.4.1 and 6.2.4.2) discussed the effect of various values for the control parameters on the performance of the ABC algorithm. The global optimum value of all test programs was (3536.5$). This section further illustrates the detail of the optimum frame obtained from all test programs. Tables (6.5) and (6.6) summarizes the optimization results for the one bay one story frame and shows the cross section dimensions as well as the reinforcement numbers and diameters. Figures (6.7) and (6.8) further illustrate these results.

Table 6. 5: Optimum column characteristics

Column Group	Width (mm)	Length (mm)	# of Bars	Diameter of Bars (mm)	Diameter of Stirrups (mm)
C1	450	500	14	22	10

Table 6. 6: Optimum beam characteristics

Beam Groups	Width (mm)	Length (mm)	Bottom Reinforcement		Upper Reinforcement			Diameter of Stirrups (mm)
			Cont.	Cut-off	Cont.	Cut-off (Left)	Cut-off (Right)	
B1	450	1450	8ø20	7ø22	3ø12	6ø20	6ø20	10

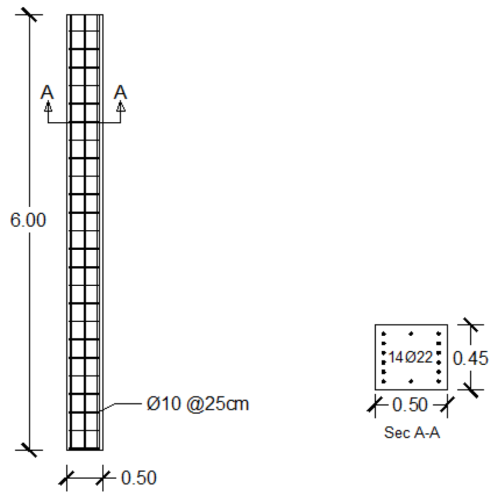


Figure 6. 7: Column results for one bay one story frame

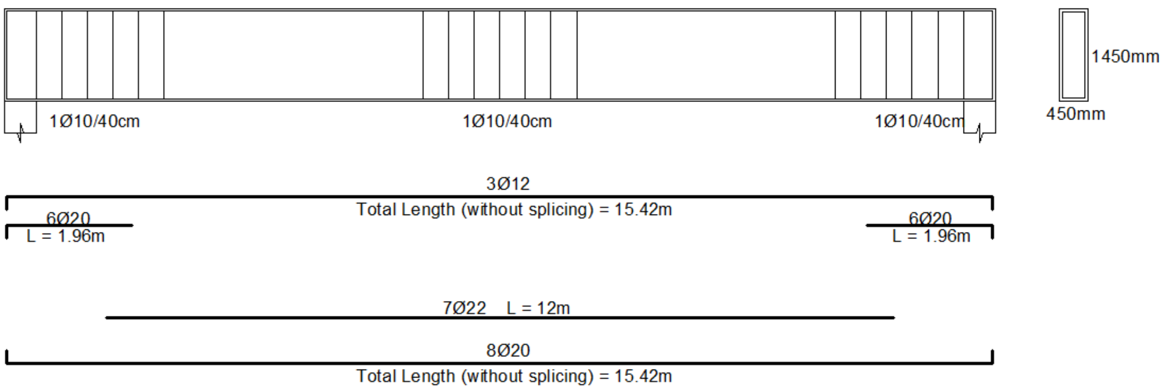


Figure 6. 8: Beam results for one bay one story frame

The optimum design utilizes the strength of its individual elements to the maximum. Figure(6.9) illustrates the strength interaction diagram of column group C1 with the most critical loading case (Dead + Live + Earthquake). The factored bending moments and axial forces are 511.62 kN.m and 720.51 kN respectively, whereas the column moment and axial strengths are 517.59 kN.m and 728.93 kN respectively. Comparing both applied loads and resisting loads, it can be concluded that the applied loads utilize 98.84% of the total resistance of the column, indicating that most of the column strength is utilized

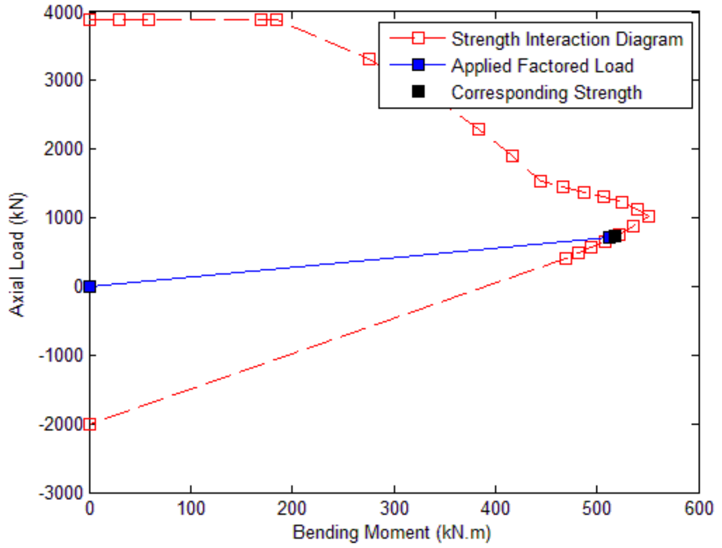


Figure 6. 9: Strength interaction diagram for column C1

6.3 Optimization of Three Bay Four Story Reinforced Concrete Frame

A three bay four story reinforced concrete framed structure, shown in Figure(6.10), is presented here to compare the optimization results obtained here versus those obtain by A. Kaveh and O. Sabzi (Kaveh & Sabzi, 2011).

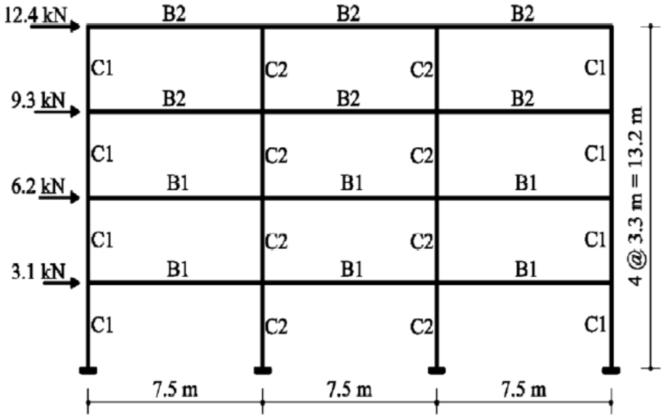


Figure 6. 10: Three bay four story reinforced concrete frame

6.3.1 Fixed Parameters of the Optimization Model

The structural inputs to model this frame in the optimization algorithm were as follows: the number of bays was set to 3, number of stories were 4, a bay length of 7.5 meters as well as a story height of 3.3 meters. The members were grouped as shown in Figure (6.10), the number of stories for column groups were 4, that is, for every 4 stories, one column group is designed based on the maximum loads, whereas the number of stories for beam groups were 2. The structure was considered symmetric and supports were set to fixed.

The loading and analysis parameters were taken identical to the frame that Kaveh and Sabzi (2011), namely: the dead load was 22.3 kN/m, the live load was 10.7 kN/m, the lateral loads were set as shown in Figure (6.10) and were considered Earthquake loads. The selfweight was not added to the dead load because the given dead load was assumed to include the selfweights of various members. No maximum permissible slenderness for columns was set in the previous study thus this study sets this parameter to 100.

For the material cost and strength, Kaveh and Sabzi used a concrete strength of 23.5 MPa, and assumed a concrete cost of 105 $\$/m^3$. For reinforcing steel, they used a yield stress of 392 MPa and a cost of 90 \$/kN. The cost of formwork was added to the total cost of the frame and was set to be 92 $\$/m^2$, whereas the cost of shear reinforcement as well as the cost for joint detailing were not considered. The unit weight of concrete was 25 kN/m^3 and for steel it was 78 kN/m^3.

The design variable limits and corresponding step sizes were given to the algorithm. These limits are required to keep the design variable values within a suitable range. Once those limits as well as their corresponding step sizes are set, one can calculate the number of possible values a design parameter can use, which is used to calculate the number of possible frame designs, also called the design space, for the design problem at hand. Table (6.7) presents the variable limits and step sizes for columns, whereas Table (6.8) shows those used for beams.

Table 6. 7: Design variables for columns (Three Bay Four Story)

Variable Name	Unit	Lower Limit	Step Size	Upper Limit	# Possible Variables Values
Column Width	mm	300	50	500	5
Column Depth	mm	250	50	900	14
Number of Reinforcing Bars used	-	4	2	20	9
Diameter of Reinforcing Bars used	mm	12	2	22	6

Table 6. 8: Design variables for beams (Three Bay Four Story)

Variable Name	Unit	Lower Limit	Step Size	Upper Limit	# Possible Variables Values
Beam Width	mm	300	50	500	5
Beam Height	mm	500	50	900	9
Number of continuous bottom reinforcement	-	2	1	10	9
Diameter of continuous bottom reinforcement	mm	12	2	22	6
Number of cutoff bottom reinforcement	-	0	1	10	11
Diameter of cutoff bottom reinforcement	mm	12	2	22	6
Number of continuous top reinforcement	-	2	1	10	9
Diameter of continuous top reinforcement	mm	12	2	22	6
Number of cutoff top reinforcement	-	0	1	10	11
Diameter of cutoff top reinforcement	mm	12	2	22	6
Diameter of Stirrups	mm	10	2	12	2

To finalize, the ABC algorithm's main control parameter had to be initialized: The number of bees taken for this problem (N_P) was 150 bees, the improvement limit for a solution (I_L) was set to 2500 trials, the Maximum number of iterations (I_{max}) was set to 5000 iterations, whereas the variable changing percentage (*VCP*) was set to 40%. Finally, the number of independent runs for this frame was set to 5 runs.

6.3.2 The Design Space

The frame presented in this section was discussed in a previous study conducted by (Kaveh & Sabzi, 2011). The number of possible frame solutions for the given possible variable combinations is calculated below:

1) Columns:
    a. Number of Possible Columns per Group = 5 x 14 x 9 x 6 x 2 = 7560 Possible Design.
    b. Number of Column Groups = 2

c. Total Number of Possible Column Designs = $7560^2 = 5.715 \times 10^7$ Possible Design

2) Beams:

a. Number of Possible Beam Designs per Group =
$5 \times 9 \times 9^2 \times 6^2 \times 11^2 \times 6^2 \times 9 \times 6 \times 11^2 \times 6^2 \times 2 = 2.689 \times 10^{14}$ Possible Design

b. Number of Beam Groups = 2

c. Total Number of Possible Beam Designs = $(2.689 \times 10^{14})^2 = 7.231 \times 10^{28}$ Possible Design

3) Total Possible Frame Designs = $(5.715 \times 10^7) \times (7.231 \times 10^{28}) = 4.13 \times 10^{36}$ Possible Frame Designs

The total number of possible designs (4.13×10^{36}) is enormous and requires a high performance optimization algorithm to deal with. Comparing this value with the value calculated for the previous frame (7.46×10^{13}) in Section (6.2.2), one can conclude that the complexity of this design optimization problem is around (5.5×10^{22}) larger.

6.3.3 Optimization Results and Discussion

The three bay four story reinforced concrete frame presented in this section, was optimized using the fixed parameters previously mentioned. This section demonstrates the results of the optimization process carried out.

6.3.3.1 Optimization Results Summary

Five independent runs were conducted and the optimum cost as well as average last improvement are tabulated in Table (6.9), followed by a summary in Table(6.10).

Table 6. 9: Results of test runs for three bay four story

Test Run	Average Last Improvement	Optimum Cost ($)
1	4643	21104
2	4488	21104
3	4925	21186
4	3230	20982
5	2754	20998

Table 6. 10: Sumary of test run results

Test Program	Average Last Improvement	Average Cost ($)	Best Cost ($)	Standard Deviation ($)
3 Bay 4Story	4008	21074.8	20982	84.53

Referring to Table (6.10), the Minimum cost of the three bay four story reinforced concrete frame is 20982$, with a standard deviation of 84.53$, or 0.4% of the Average Cost of 21074.8$. Such results are satisfying taking into consideration the size of the design pool discussed in section (6.3.2).

The convergence history of the best test run (test run 4) is shown in Figure(6.11). The figure shows a typical convergence history that starts with a very steep slope then goes into a somehow steady state characterized by its low slope. Figure (6.12) focuses on the first 500 iterations to further illustrate the transition from steep slope to steady state. Comparing the results obtained here with the one bay one story frame results obtained in section (6.2.4), it can be concluded that the convergence rate of this frame is slower, which can be related to the significant increase in the design space..

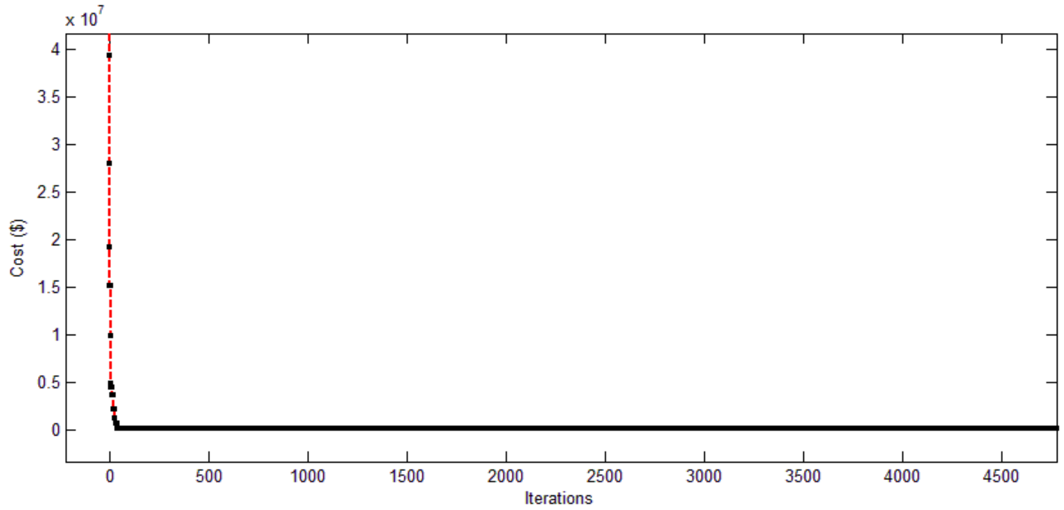


Figure 6. 11: Convergence history of optimization of three bay four story frame

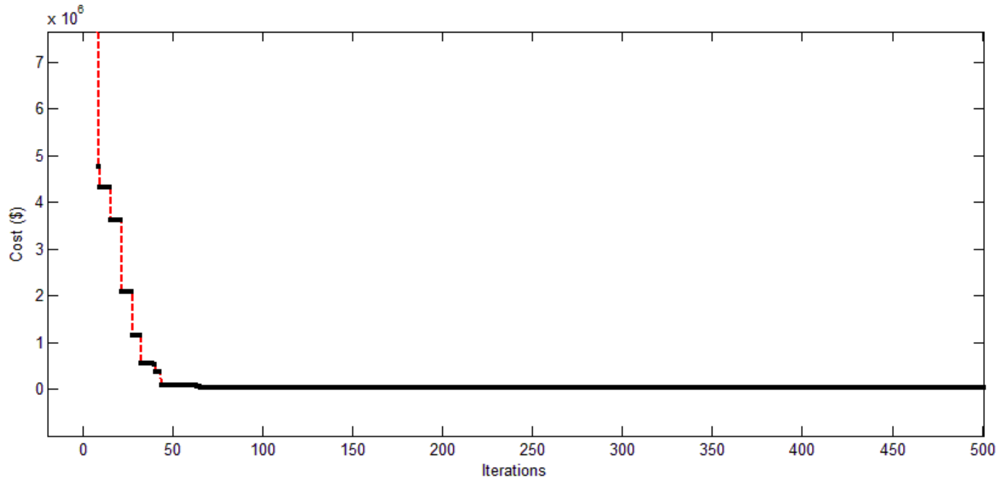


Figure 6. 12: Transition from steep slope to steady state for three bay four story frame

The best cost at the 500th iteration was 23109.84$, which is 10% larger than the optimum cost of 20982$. However, at the 1000th iteration, which is considered 20% of the total number of iterations, the cost was 22290.45$, which is 6.2% larger than the optimum cost of 20982$. The convergence may seem slow, but still acceptable when the massiveness of the design space is considered.

6.3.3.2 Optimum Solution Details

The previous section (6.3.3.1) discussed the summary of the optimizations results for a three bay four story frame. The global optimum value of all test runs was 20982$. This section further illustrates the detail of the optimum frame obtained from all test runs.

Tables (6.11) and (6.12) summarizes the optimization results for the three bay four story frame and shows the cross section dimensions as well as the reinforcement numbers and diameters. Figures (6.13) through (6.16) further illustrate these results.

Table 6. 11: Optimum column characteristics

Column Group	Width (mm)	Length (mm)	# of Bars	Diameter of Bars (mm)	Diameter of Stirrups (mm)
C1	300	350	8	22	10
C2	300	350	12	12	10

Table 6. 12: Optimum beam characteristics

Beam Groups	Span	Width (mm)	Height (mm)	Bottom Reinforcement		Upper Reinforcement			Diameter of Stirrups (mm)
				Cont.	Cut-off	Cont.	Cut-off (Left)	Cut-off (Right)	
B1	1	300	500	4ø16	-	4ø14	3ø18	3ø22	10
	2			2ø22	-		3ø22	3ø22	
	3			4ø16	-		3ø22	3ø18	
B2	1	300	500	3ø20	-	3ø12	3ø22	4ø22	10
	2			2ø22	-		4ø22	4ø22	
	3			3ø20	-		4ø22	3ø22	

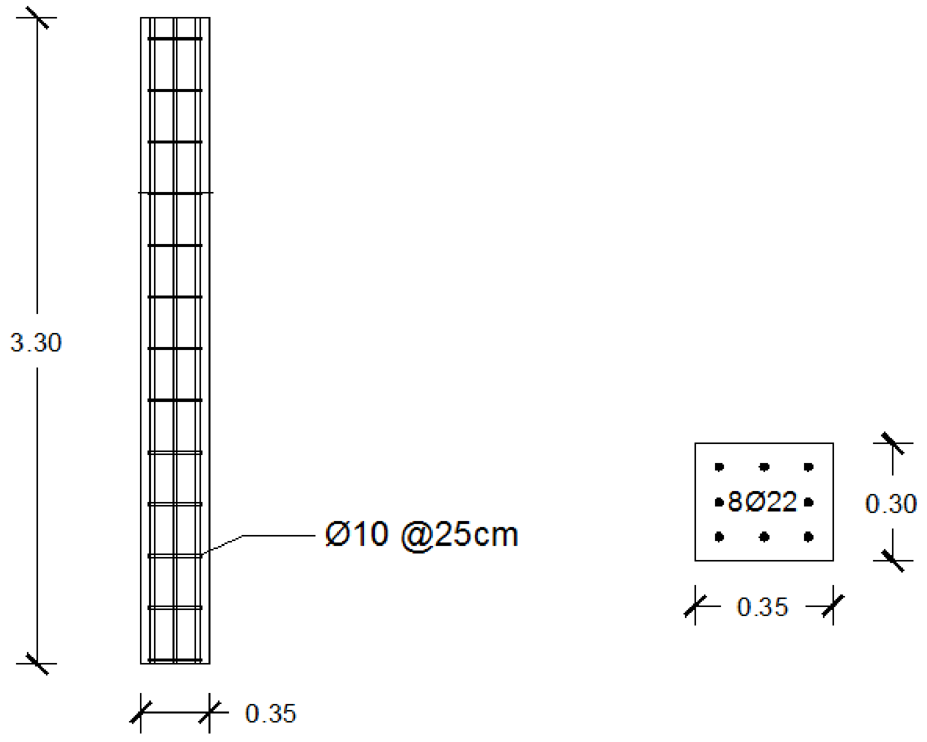


Figure 6. 13: Column results for three bay four story frame (Group C1)

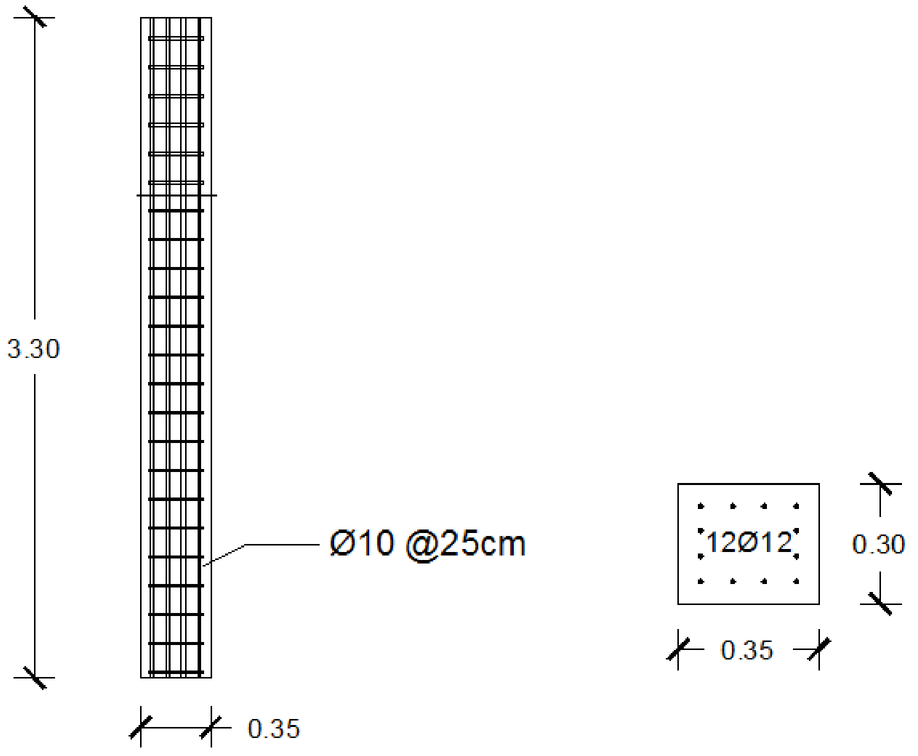


Figure 6. 14: Column results for three bay four story frame (Group C2)

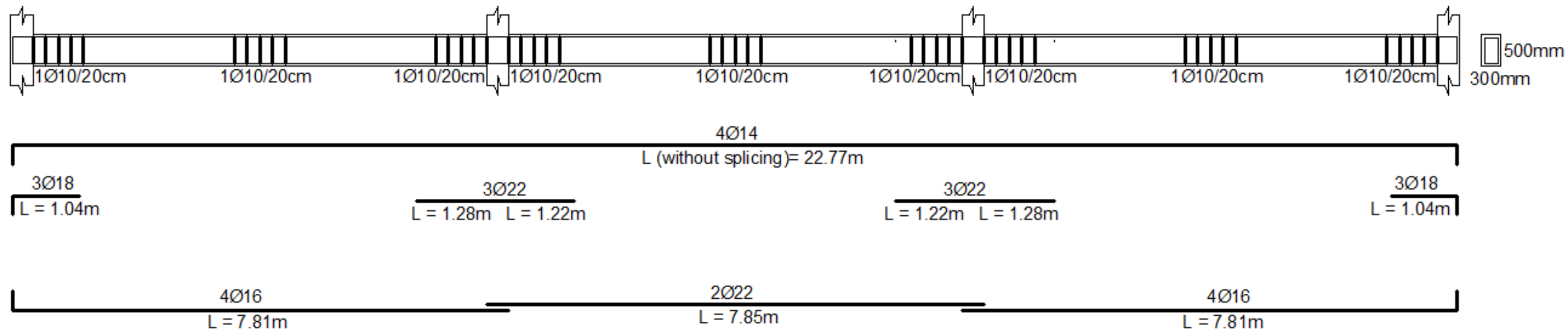


Figure 6. 15: Beam results for three bay four story frame (Group B1)

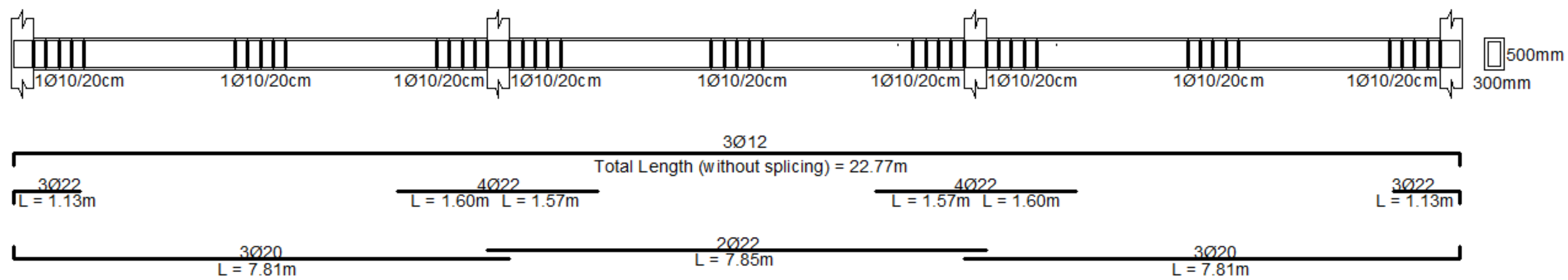


Figure 6. 16: Beam results for three bay four story frame (Group B2)

Once again, the optimum design succeeded in utilizing the strength of its individual elements to the maximum. To illustrate this, consider Figure (6.17) which shows the strength interaction diagram for the most critical column in group C1, which is located in the fourth floor, and is subjected to the loading case (Dead + Live). The factored bending moments and axial forces are 140.49 kN.m and 152.97 kN respectively, whereas the column moment and axial strengths are 141.47 kN.m and 154.03 kN respectively. Comparing both applied loads and resisting loads, it can be concluded that the applied loads utilize 99.3% of the total resistance of column group C1.

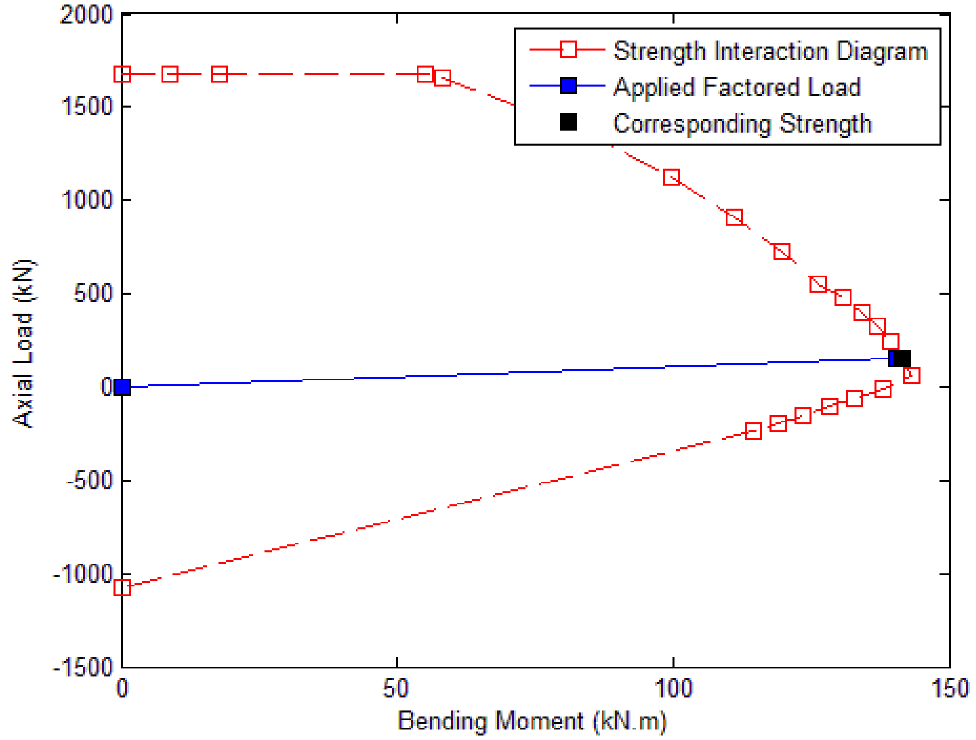


Figure 6. 17: Strength interaction diagram for column (C1)

Observing Figure (6.17), which shows the strength interaction diagram of column group C1, one can conclude that the loads which controlled the design of column group C1 are characterized by large moments and small axial forces, which is normal since the most critical column in column group C1 was located in the fourth floor. It can also be concluded that the design of edge column groups, which are connected to beams from one side only, are controlled by columns located in higher stories, where the axial forces decrease, decreasing the overall capacity of columns.

Figure (6.18) illustrates the strength interaction diagram of the most critical column in column group C2, which is located in the first floor, and is subjected to the loading case (Dead + Live), too. The factored bending moments and axial forces are 5.64 kN.m and

1345.6 kN respectively, whereas the column moment and axial strengths are 5.67 kN.m and 1353.18 kN respectively. Once again, it can be concluded that the applied loads utilize 99.4% of the total resistance of column group C2.

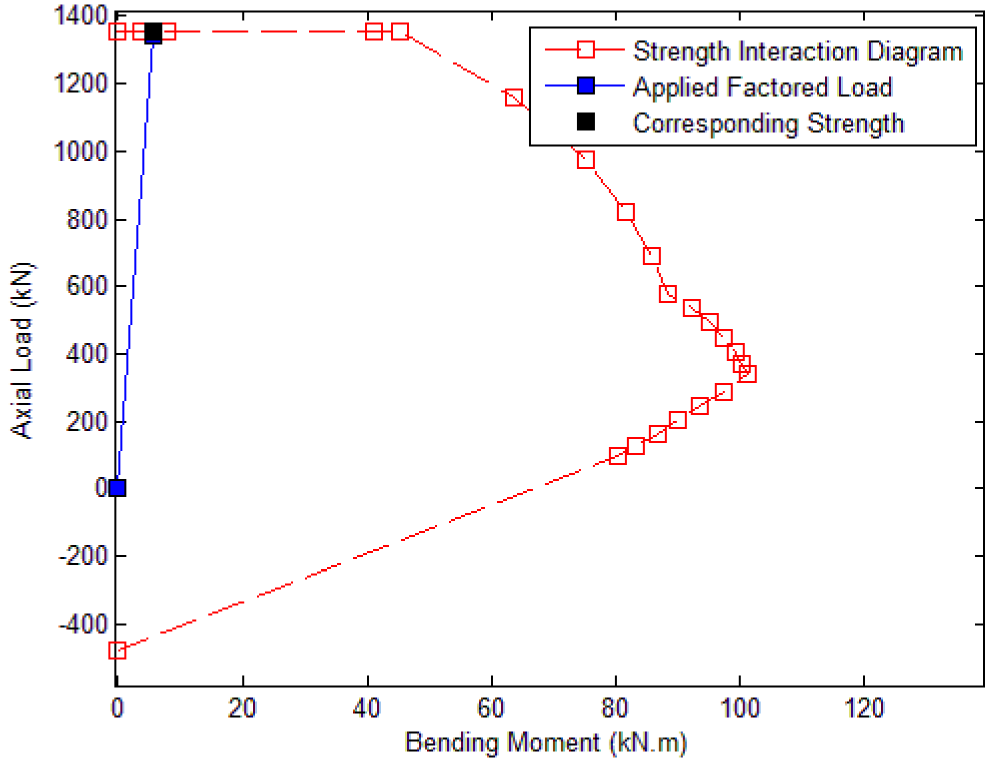


Figure 6. 18: Strength interaction diagram for column (C2)

Observing Figure (6.18), which shows the strength interaction diagram of column group C2, one can conclude that the loads that controlled the design of column group C2 are characterized by low moments and high axial forces, which is normal since the most critical column in column group C2 was located in the first floor. One can conclude that the design of interior column groups, which are connected to beams from both sides, are controlled by columns located in lower stories, where the axial forces increase, since the bending moments are kept to minimum due to a balance in loading on both sides.

6.3.3.3 Comparison of optimization results with previous studies

The three bay four story reinforced concrete frame presented in this section was previously studied by Kaveh and Sabzi (2011), using two optimization algorithms. Both optimization algorithms were a combination of various smaller algorithms. The first one was a heuristic big bang-big crunch (HBB-BC), which is based on big bang-big crunch (BB-BC) and a harmony search (HS) scheme to deal with the variable constraint. The second one is The (HPSACO) algorithm, which is a combination of particle swarm with passive congregation (PSOPC), ant colony optimization (ACO), and harmony search scheme (HS) algorithms.

The design variables used by Kaveh and Sabzi were much simpler than those used in this study and were limited to the section dimensions as well as the number of reinforcing bars and their diameter without using cut-off bars. Such a simplification significantly decreases the design pool making optimization simpler. However, such a simplification increases the total cost significantly. Furthermore, reinforced concrete frames are usually designed utilizing cut-off bars.

Table (6.13), taken from (Kaveh & Sabzi, 2011), shows the optimization results of the frame discussed in this section using two different optimization technique. The ABC algorithm and design optimization technique presented in this study obtained a cost of 20982$ using the same cost, material and load parameters. This cost is 5.5% cheaper than the 22207$ cost obtained by (Kaveh & Sabzi, 2011) which is shown in Table (6.13).

This proves the superiority of the ABC Algorithm and the design approach considered in the research work of this thesis in both performance and robustness when compared with the (HSPACO and (HBB-BC) approaches used in (Kaveh & Sabzi, 2011).

Table 6. 13: Design results for three bay four story frame (Kaveh & Sabzi 2011)

		HPSACO				HBB-BC			
		Sectional Dimensions		Reinforcements		Sectional Dimensions		Reinforcement	
Member type	Element group	Width (mm)	Depth (mm)	Positive moment	Negative moment	Width (mm)	Depth (mm)	Positive moment	Negative moment
Beam	B1	300	500	3ø19	5ø22	300	500	3ø19	5ø22
	B2	300	500	4ø19	5ø22	300	500	4ø19	5ø22
Column	C1	350	350	8ø25		350	350	8ø25	
	C2	300	300	6ø25		300	300	6ø25	
Frame Cost		22207 $				22207 $			

It can be concluded that the utilization of cut off bars as well as the excellent performance of the ABC algorithm in engineering optimization, led to extra savings in the overall cost of reinforced concrete frames, which is around 5.5% for the frame studied and optimized in this section.

6.4 Optimization of Three Bay Eight Story Reinforced Concrete Frame

A three bay eight story reinforced concrete framed structure, shown in Figure(6.19), is presented here to further push the ABC algorithm to its limits. The frame presented in this section is of significant complexity in both analysis and design and requires a high performance algorithm to deal with. The results obtained using the ABC are compared with those obtain by A. Kaveh and O. Sabzi (Kaveh & Sabzi, 2011).

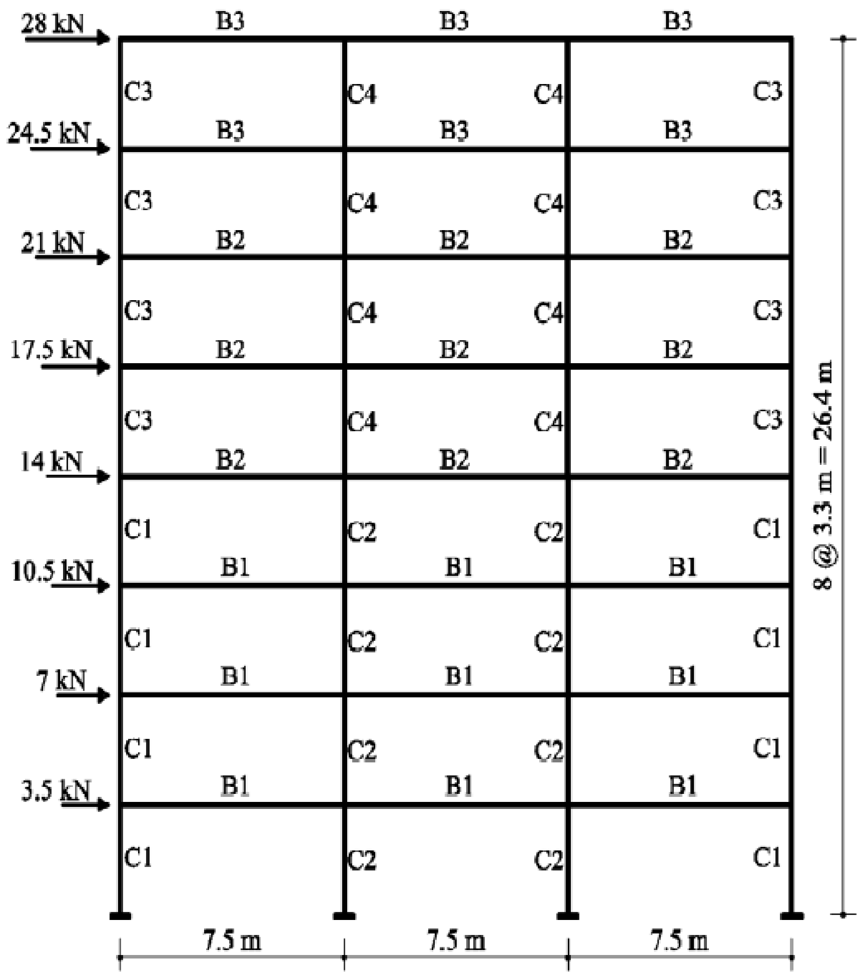


Figure 6. 19: Three bay eight story reinforced concrete frame

6.4.1 Fixed Parameters of the Optimization Model

The structural inputs to model this frame in the optimization algorithm were as follows: the number of bays was set to 3, number of stories were 8, a bay length of 7.5 meters as well as a story height of 3.3 meters. The members were grouped as shown in Figure (6.19), the number of stories for column groups were 4, whereas the number of stories for beam groups was set to 3. The structure was considered symmetric and supports were set to fixed.

The loading and analysis parameters were taken identical to the frame that Kaveh and Sabzi (2011), namely: the dead load was 22.3 kN/m, the live load was 10.7 kN/m, the lateral loads were set as shown in Figure (6.19) and were considered Earthquake loads. The selfweight was not added to the dead load because the given dead load was assumed to include the selfweights of various members. No maximum permissible slenderness for columns was set in the previous study thus this study sets this parameter to 100.

For the material cost and strength, Kaveh and Sabzi used a concrete strength of 23.5 MPa, and assumed a concrete cost of 105 $\$/m^3$. For reinforcing steel, they used a yield stress of 392 MPa and a cost of 90 $/kN. The cost of formwork was added to the total cost of the frame and was set to be 92 $/m^2, whereas the cost of shear reinforcement as well as the cost for joint detailing were not considered.

The variable limits, step sizes and possible values are shown below. Table (6.14) presents the variable limits and step sizes for beams, whereas Table (6.15) shows those used for columns.

Table 6. 14: Design variables for beams (Three Bay Eight Story)

Variable Name	Unit	Lower Limit	Step Size	Upper Limit	# Possible Variables Values
Beam Width	mm	300	50	500	5
Beam Height	mm	500	50	900	9
Number of continuous bottom reinforcement	-	2	1	5	4
Diameter of continuous bottom reinforcement	mm	12	2	22	6
Number of cutoff bottom reinforcement	-	0	1	5	6
Diameter of cutoff bottom reinforcement	mm	12	2	22	6
Number of continuous top reinforcement	-	2	1	5	4
Diameter of continuous top reinforcement	mm	12	2	22	6
Number of cutoff top reinforcement	-	0	1	5	6
Diameter of cutoff top reinforcement	mm	12	2	22	6
Diameter of Stirrups	mm	10	2	12	2

Table 6. 15: Design variables for columns (Three Bay Eight Story)

Variable Name	Unit	Lower Limit	Step Size	Upper Limit	# Possible Variables Values
Column Width	mm	300	50	500	5
Column Depth	mm	250	50	900	14
Number of Reinforcing Bars used	-	4	2	20	9
Diameter of Reinforcement	mm	12	2	22	6

To finalize, the ABC algorithm's main control parameter had to be initialized: The number of bees taken for this problem (N_P) was 150 bees, the improvement limit for a solution (I_L) was set to 7500 trials, the Maximum number of iterations (I_{max}) was set to 5000 iterations, whereas the variable changing percentage (*VCP*) was set to 40%. Finally, the number of independent runs for this frame was set to 5 runs.

6.4.2 The Design Space

The frame presented in this section was discussed in a previous study conducted by (Kaveh & Sabzi, 2011). The number of possible frame solutions for the given possible variable combinations is calculated below:

1) Columns:
    a. Number of Possible Columns per Group = 5 x 14 x 9 x 6 x 2 = 7560 Possible Design.
    b. Number of Column Groups = 4
    c. Total Number of Possible Column Designs = 7560^4 = 3.266 x 10^{15} Possible Design
2) Beams:
    a. Number of Possible Beam Designs per Group = 5 x 9 x 4^2 x 6^2 x 6^2 x 6^2 x 4 x 6 x 6^2 x 6^2 x 2 = 2.089 x 10^{12} Possible Design
    b. Number of Beam Groups = 2
    c. Total Number of Possible Beam Designs = $(2.089 \times 10^{12})^3$ = 9.125 x 10^{36} Possible Design
3) Total Possible Frame Designs = (3.266 x 10^{15}) x (9.125 x 10^{36}) = 2.98 x 10^{52} Possible Frame Designs

The total number of possible designs (2.98 x 10^{52}) pushes the ABC algorithm's performance and robustness to the ultimate limit. Comparing this value with the value calculated for the previous three bay four story frame (4.13 x 10^{36}) in Section (6.3.2), one can conclude that the complexity of this design optimization problem is around (7.2 x 10^{15}) larger.

6.4.3 Optimization Results and Discussion

The three bay eight story reinforced concrete frame presented in this section, was optimized using the fixed parameters previously mentioned. This section demonstrates the results of the optimization process carried out.

6.4.3.1 Optimization Results Summary

Five independent runs were conducted and the results are tabulated in Table (6.16), followed by a summary in Table(6.17).

Table 6. 16: Results of test runs for three bay eight story frame

Test Run	Average Last Improvement	Optimum Cost ($)
1	3663	47514
2	4215	46800
3	4540	47381
4	3230	46800
5	4011	47381

Table 6. 17: Summary of test run results

Test Program	Average Last Improvement	Average Cost ($)	Best Cost ($)	Standard Deviation ($)
3 Bay 8Story	3931.8	47175.2	46800	346.78

Referring to Table (6.17), the Minimum cost of the three bay eight story reinforced concrete frame is 46800$, with a standard deviation of 346.78$, or 0.7% of the Average Cost of 47175.2$. Such results are relatively good taking into consideration the size of the design pool discussed in section (6.4.2).

The convergence history of the best test run (test run 4) is shown in Figure(6.20). Whereas Figure (6.21) focuses on the first 500 iterations to further illustrate the transition from steep slope to steady state.

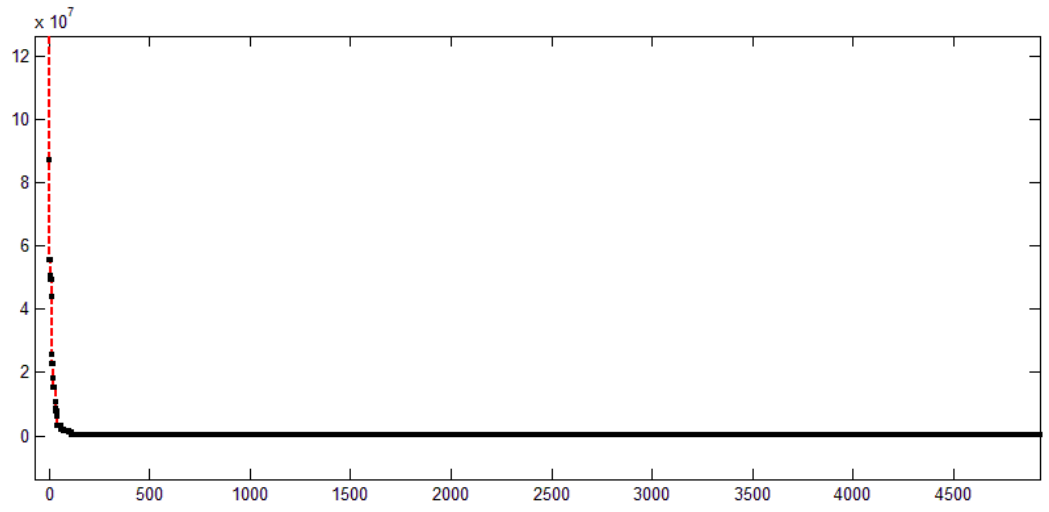


Figure 6. 20: Convergence history of optimization of three bay eight story frame

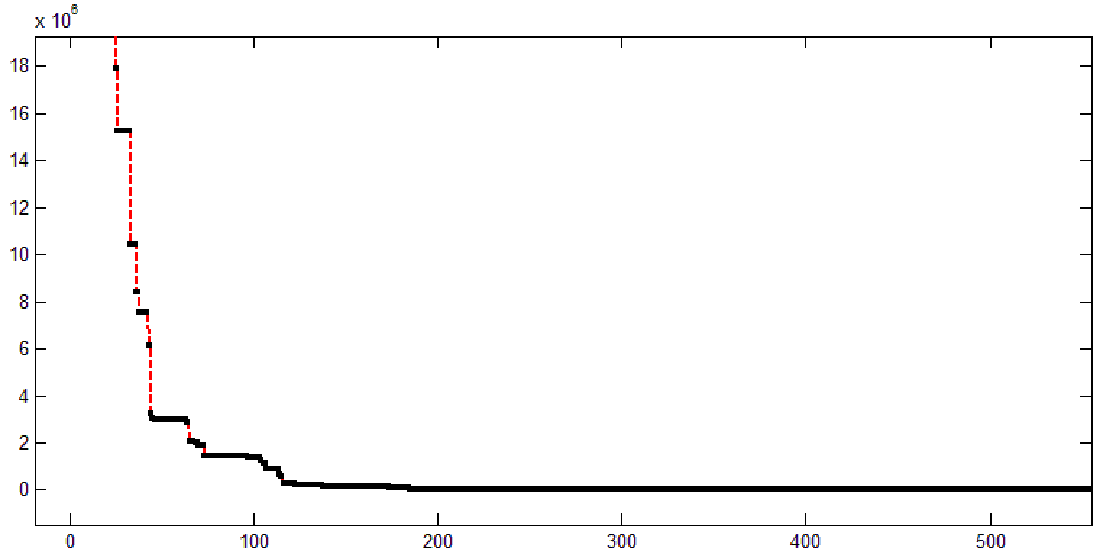


Figure 6. 21: Transition from steep slope to steady state for three bay eight story frame

The best cost at the 500th iteration was 51274.37$, which is 9.5% larger than the optimum cost of 46800$. However, at the 1000th iteration, which is considered 20% of the total number of iterations, the cost was 48963.53$, which is 4.6% larger than the optimum cost of 46800$. The convergence may seem slow, but still acceptable when the massiveness of the design space is considered.

6.4.3.2 Optimum Solution Details

The previous section (6.4.3.1) discussed summary of the optimizations results for a three bay eight story frame. The global optimum value of all test runs was 46800$. This section further illustrates the detail of the optimum frame elements and reinforcements obtained from all test runs.

Tables (6.18) and (6.19) summarizes the optimization results for the three bay eight story frame and shows the cross section dimensions as well as the reinforcement numbers and diameters. Figures (6.22) through (6.28) further illustrate these results.

Table 6. 18: Optimum column characteristics

Column Group	Width (mm)	Length (mm)	# of Bars	Diameter of Bars (mm)	Diameter of Stirrups (mm)
C1	350	500	12	14	10
C2	350	550	12	20	10
C3	300	400	10	18	10
C4	300	350	10	14	10

Table 6. 19: Optimum beam characteristics

Beam Groups	Span	Width (mm)	Height (mm)	Bottom Reinforcement		Upper Reinforcement			Diameter of Stirrups (mm)
				Cont.	Cut-off	Cont.	Cut-off (Left)	Cut-off (Right)	
B1	1	300	600	4ø16	-	3ø12	4ø22	4ø22	10
	2			5ø12	-		4ø22	4ø22	
	3			4ø16	-		4ø22	4ø22	
B2	1	300	500	5ø14	-	3ø14	4ø22	4ø22	10
	2			5ø12	2ø12		4ø22	4ø22	
	3			5ø14	-		4ø22	4ø22	
B3	1	300	500	4ø12	4ø12	3ø16	3ø22	4ø20	10
	2			3ø14	2ø14		4ø20	4ø20	
	3			4ø12	4ø12		4ø20	3ø22	

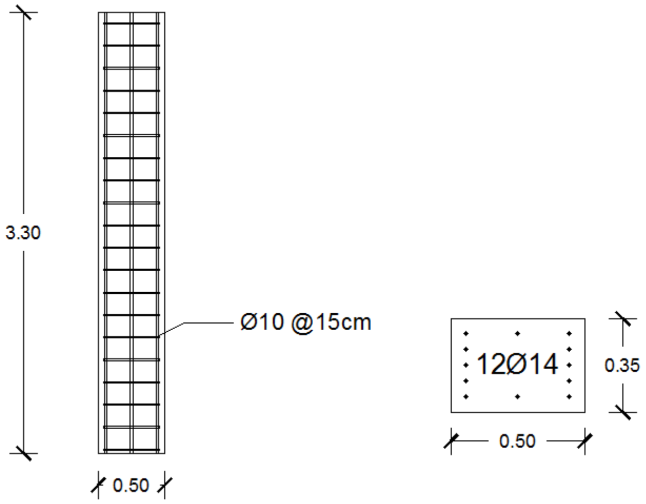


Figure 6. 22: Column results for three bay eight story frame (Group C1)

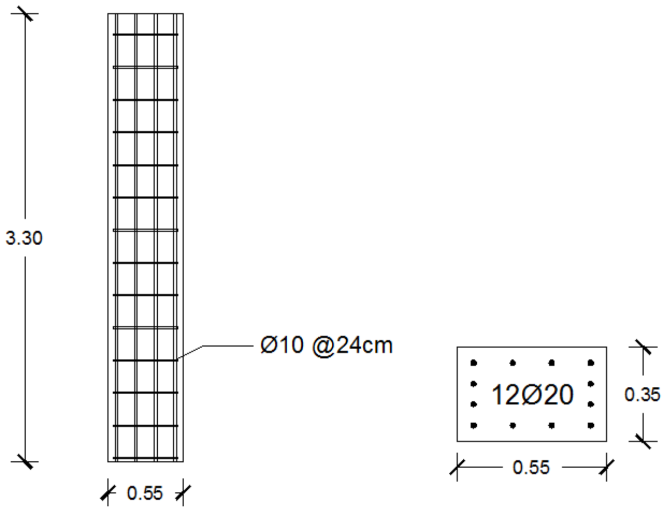


Figure 6. 23: Column results for three bay eight story frame (Group C2)

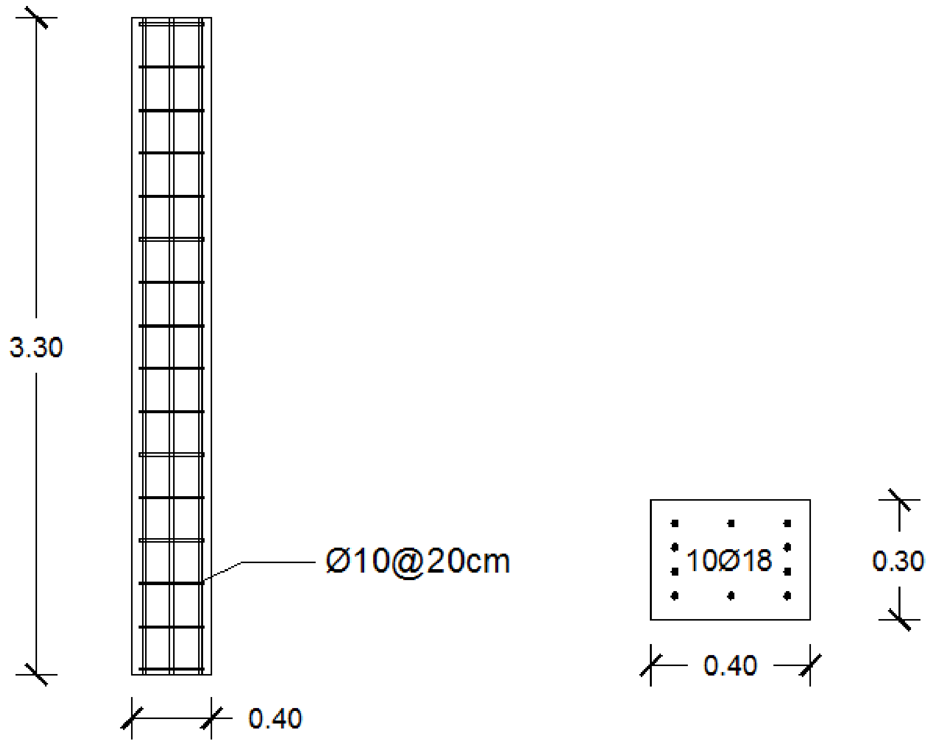


Figure 6. 24: Column results for three bay eight story frame (Group C3)

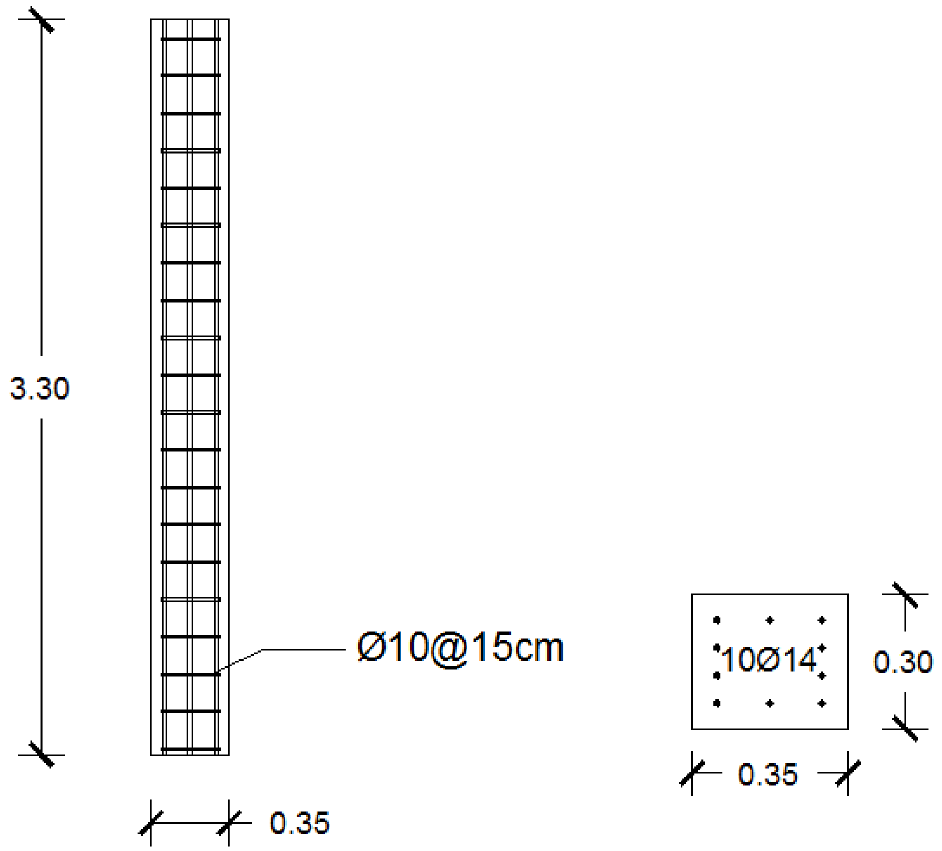


Figure 6. 25: Column results for three bay eight story frame (Group C4)

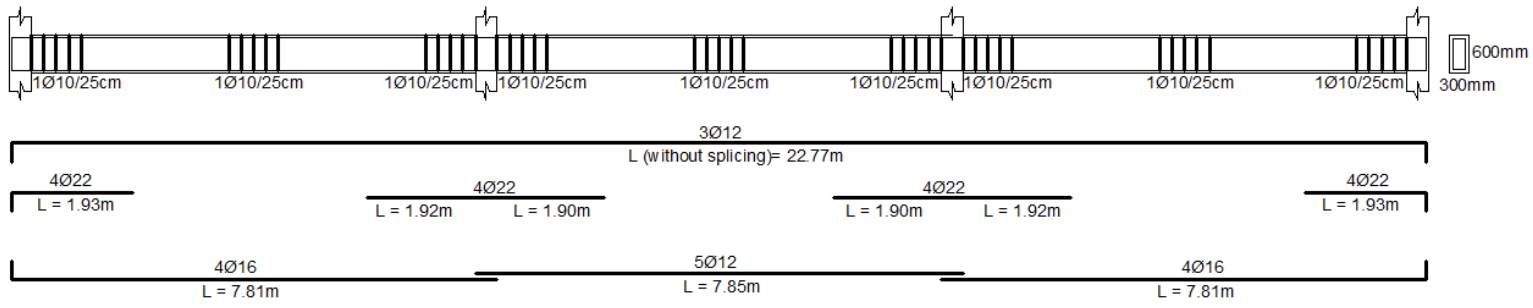


Figure 6. 26: Beam results for three bay eight story frame (Group B1)

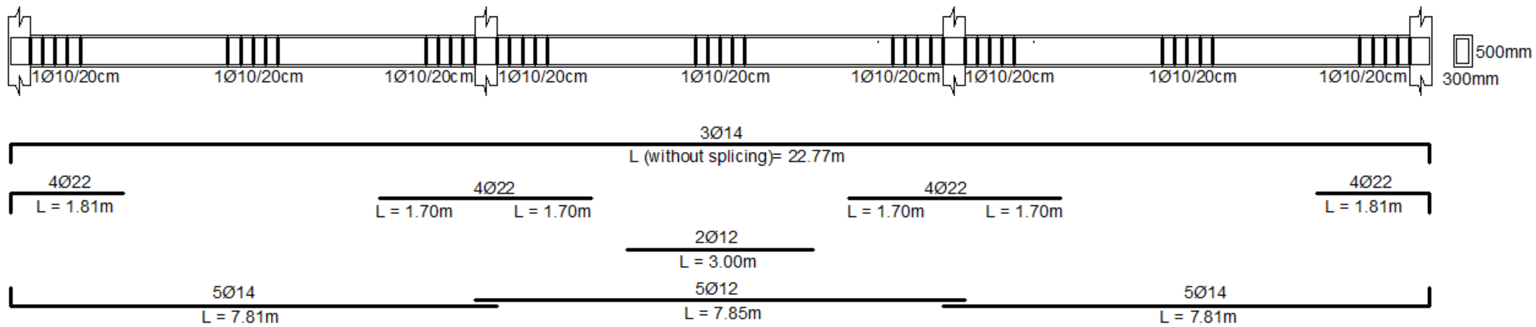


Figure 6. 27: Beam results for three bay eight story frame (Group B2)

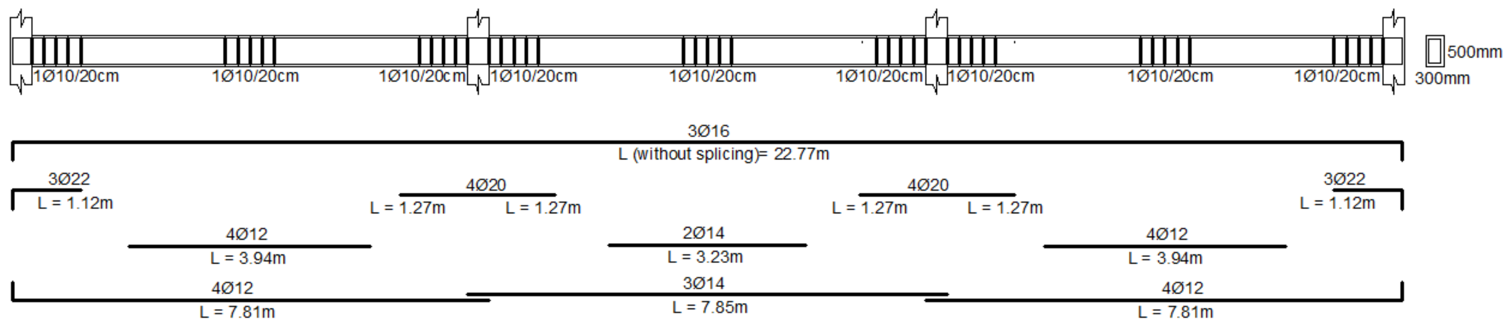


Figure 6. 28: Beam results for three bay eight story frame (Group B3)

The optimum design utilizes the strength of its individual elements to the maximum. To illustrate this, consider Figure(6.29) which illustrates the strength interaction diagram of the most critical column in column group C1, which is located in the first floor, and is subjected to the loading case (Dead + Live + Earthquake). The factored bending moments and axial forces are 153.64 kN.m and 1078.16 kN respectively, whereas the column moment and axial strengths are 199.06 kN.m and 1396.88 kN respectively. Comparing both applied loads and resisting loads, it can be concluded that the applied loads utilize 77.2% of the total resistance of column group C1, which is rather low. The reason for such a low utilization of column strength could be due to other constraints, such as lateral deflection. Another reason could be the minimum slenderness ratio or the importance of column rigidity in order to decrease beam moments.

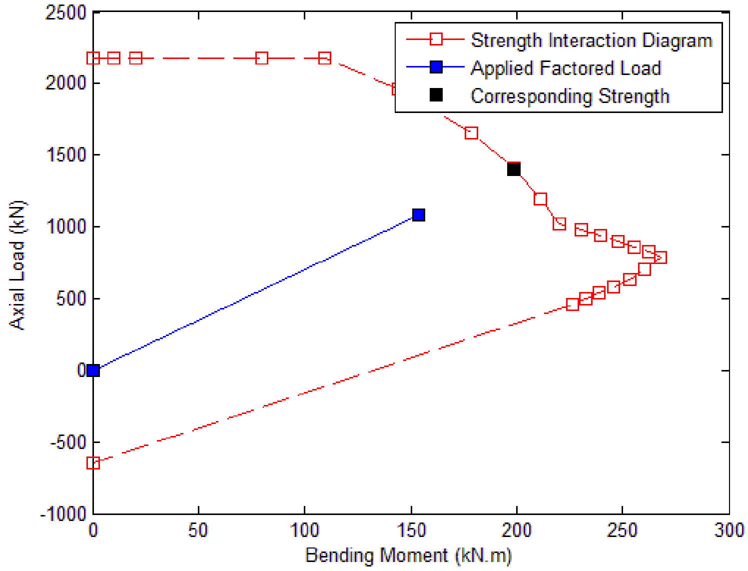


Figure 6. 29: Strength interaction diagram for column (C1)

Figure (6.30) illustrates the strength interaction diagram of the most critical column in column group C2, which is located in the first floor, and is subjected to the loading case (Dead + Live). The factored bending moments and axial forces are 2.48 kN.m and 2653.72 kN respectively, whereas the column moment and axial strengths are 2.55 kN.m and 2728.8 kN respectively. Comparing both applied loads and resisting loads, it can be concluded that the applied loads utilize 97.2% of the total resistance of column group C2.

Once again, the design of interior column group C2 was controlled by large axial forces and low moments, since moments are low in interior columns whereas axial forces are high. Such loading indicate the most critical column in column group C2 should be located in the first floor, which was found to be so.

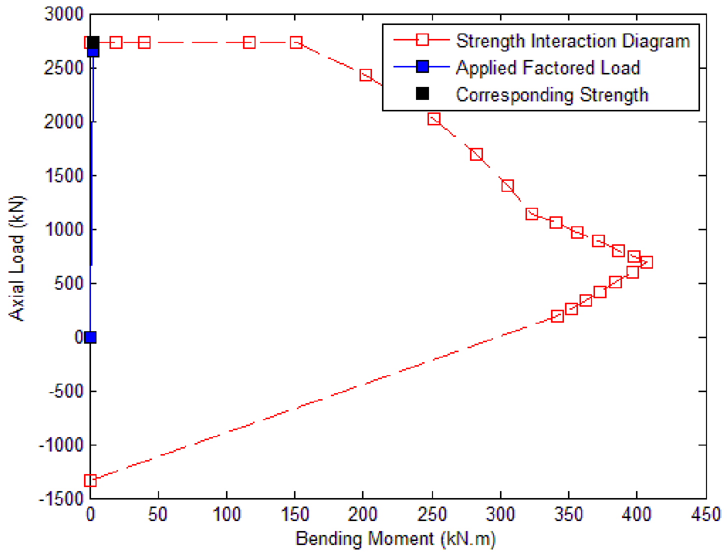


Figure 6. 30: Strength interaction diagram for column (C2)

Figure (6.31) illustrates the strength interaction diagram of the most critical column in column group C3, which is located in the eighth floor, and is subjected to the loading case Dead + Live. The factored bending moments and axial forces are 161.97 kN.m and 157.34 kN respectively, whereas the column moment and axial strengths are 164.7 kN.m and 159.99 kN respectively. Comparing both applied loads and resisting loads, it can be concluded that the applied loads utilize 98.3% of the total resistance of column group C3.

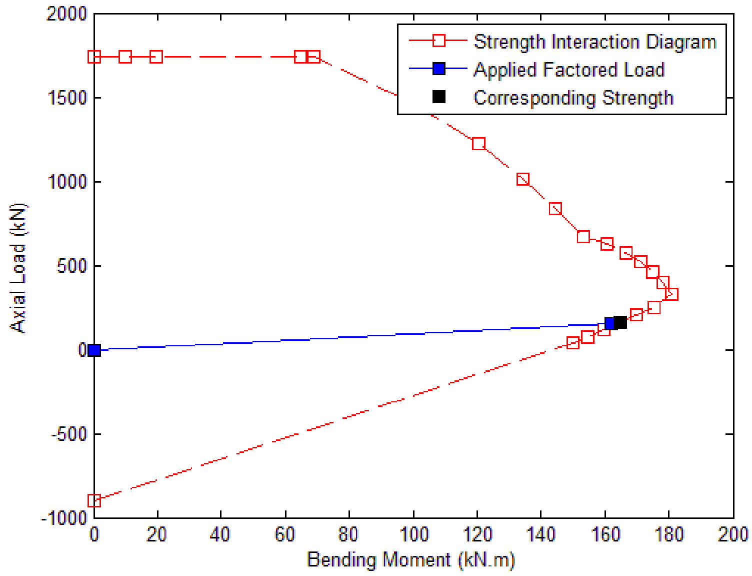


Figure 6. 31: Strength interaction diagram for column (C3)

Observing Figure (6.31), which shows the strength interaction diagram of column group C3, one can conclude that the loads that controlled the design of column group C3 are characterized by large moments and small axial forces, which is normal since the most critical column in column group C3 was located in the eighth floor. One can conclude that the design of edge column groups, which are connected to beams from one side only, are controlled by columns located in higher stories, where the axial forces decrease, decreasing the overall capacity of columns.

Figure (6.32) illustrates the strength interaction diagram of the most critical column in column group C4, which is located in the fifth floor, and is subjected to the loading case (Dead + Live + Earthquake). The factored bending moments and axial forces are 69.4 kN.m and 1141.66 kN respectively, whereas the column moment and axial strengths are 69.78 kN.m and 1147.91 kN respectively. Comparing both applied loads and resisting loads, it can be concluded that the applied loads utilize 99.4% of the total resistance of column group C4.

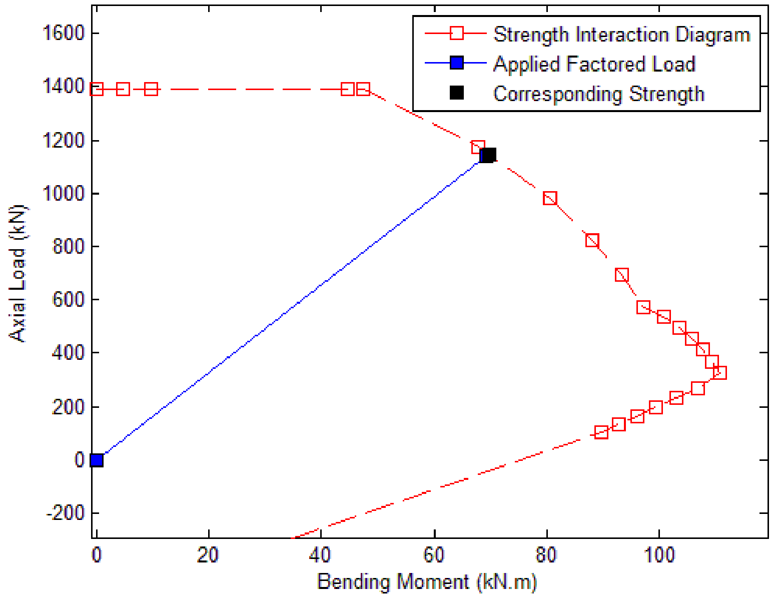


Figure 6. 32: Strength interaction diagram for column (C4)

6.4.3.3 Comparison of optimization results with previous studies

The ABC algorithm as well as the design variables used in this study, once again, proved its efficiency in dealing with the three bay eight story reinforced concrete frame when compared with the findings Kaveh and Sabzi (2011) obtained. Table (6.20), taken from (Kaveh & Sabzi, 2011), shows the optimization results for the frame discussed in this section using two different optimization technique. The ABC algorithm and design optimization technique presented in the research works of this thesis obtained a cost of 46800$ using the same cost, material and load parameters. This cost is 3.5% cheaper than the cost obtained by (Kaveh & Sabzi, 2011) using HPSACO algorithm, and 3.0% cheaper than that using the HBB-BC algorithm. The costs obtained by Kaveh and Sabzi are shown in Table (6.20).

This proves the superiority of the ABC Algorithm and the design approach considered in the research work of this thesis in both performance and robustness when compared with the (HSPACO) and (HBB-BC) approaches used in (Kaveh & Sabzi, 2011), even if infinite search spaces are involved.

Table 6. 20: Design results for three bay eight story frame (Kaveh & Sabzi 2011)

		HPSACO				HBB-BC			
		Sectional Dimensions		Reinforcements		Sectional Dimensions		Reinforcement	
Member type	Element group	Width (mm)	Depth (mm)	Positive moment	Negative moment	Width (mm)	Depth (mm)	Positive moment	Negative moment
Beam	B1	300	500	3ø19	6ø22	300	500	3ø19	6ø22
	B2	300	500	3ø19	6ø22	300	500	3ø19	6ø22
	B3	300	500	3ø19	5ø22	300	500	3ø19	5ø22
Column	C1	400	400	8ø25		400	400	8ø25	
	C2	500	500	8ø25		450	450	12ø25	
	C3	350	350	8ø25		350	350	8ø25	
	C4	350	350	8ø25		350	350	8ø25	
Frame Cost		48514 $				48263 $			

6.5 Concluding Remarks

Three reinforced concrete frames were successfully optimized using the Artificial Bee Colony (ABC) Algorithm in this chapter, with two of them previously studied by (Kaveh & Sabzi, 2011). The results reflect the robustness and performance of the ABC algorithm when compared to the optimization algorithms used by (Kaveh & Sabzi, 2011).

The results for the three bay four story frame and three bay eight story were compared with a study previously conducted by Kaveh and Sabzi (2011) using two different optimization algorithms: The heuristic big bang-big crunch (HBB-BC), which is based on big bang-big crunch (BB-BC) and a harmony search (HS) scheme to deal with the variable constraint, and The (HPSACO) algorithm, which is a combination of particle swarm with passive congregation (PSOPC), ant colony optimization (ACO), and harmony search scheme (HS) algorithms. The results prove that the ABC algorithm as well as the design variables used in this thesis yield better results than the previous study: For the three bay four story frame, savings of (5.5%) were achieved whereas for the three bay, eight story frame, savings of (3.0%) were achieved.

CHAPTER 7: CONCLUSIONS AND FUTURE RESEARCH

7.1 Introduction

The main objective of the current study was to develop an optimization model that is capable of obtaining the optimum design for reinforced concrete frames in terms of cross section dimensions and reinforcement details. The optimization was carried out using Artificial Bee Colony (ABC) Algorithm, while still satisfying the strength and serviceability constraints of the American Concrete Institute Building Code Requirements for Structural Concrete and Commentary (ACI318M-08), this model is then applied to study cases to obtain results and draw possible conclusions and recommendations.

Three different reinforced concrete frames were subject to optimization, with the first one acting as a test frame to obtained the best combination of control factors for the ABC algorithm. The other two frames were selected from previous studies for comparison purposes. This chapter briefly discusses the conclusions drawn from the research work of this thesis and suggests areas of future research for the reader to follow.

7.2 Conclusions

The use of the Artificial Bee Colony Algorithm in the optimization of reinforced concrete frames conforming to the ACI318-08 code has been achieved. The conclusions can be summarized as follows:

- General Conclusions about the ABC Algorithm and Design Variables used:
  a. The original ABC Algorithm cannot deal efficiently with the design of reinforced concrete members, since a member is described using multiple variables. Thus, a modification on the original algorithm had to be done, which was achieved by introducing a new control variable called: The Variable Changing Percentage (*VCP*).
  b. The ABC Algorithm is a high performance algorithm: it dealt with extreme design spaces as large as 2.98 x 10^{52} successfully and gave reliable results.
  c. The ABC Algorithm proved to be a rapid converging algorithm, always obtaining near optimal solutions within fraction of the total number of iterations available for a single run
  d. The ABC Algorithm proved to be a reliable and robust algorithm: No matter how large the design space, the standard deviation of various test runs was always kept to a minimum
  e. Design optimization using broader design variables certainly results in cost savings, but still increases the complexity of the problem. A researcher must practice extreme care neither to overcomplicate the design problem nor to oversimplify it.

- Conclusion drawn from the optimization of one bay one story reinforced concrete frame:
    a. The frame was successfully optimized and reliable results with minimum deviation were obtained
    b. The best performance of the ABC algorithm was obtained when the number of bees (N_P) was set to 100 and the variable changing percentage (*VCP*) was taken equal to 40%.
- Conclusions drawn from the optimization of three bay four story reinforced concrete frame:
    a. The ABC Algorithm achieved (5.5%) cost savings for the three bay-four story frame when compared with (Kaveh & Sabzi, 2011). The frame was presented previously in the research work of this thesis
    b. The algorithm succeeded in dealing with a design space of 4.13×10^{36} with a relatively small deviation of 0.4%
    c. The design of edge column groups, which are connected to beams from one side only, are controlled by columns located in higher stories, where the axial forces decrease, decreasing the overall capacity of columns.
    d. The design of interior column groups, which are connected to beams from both sides, are controlled by columns located in lower stories, where the axial forces increase, since the bending moments are kept to minimum due to a balance in loading on both sides.
- Conclusions drawn from the optimization of three bay eight story reinforced concrete frame:
    a. The ABC Algorithm achieved (3.5%) cost savings for the three bay-eight story frame when compared with (Kaveh & Sabzi, 2011). The frame was also presented previously in the research work of this thesis.
    b. The algorithm succeeded in dealing with a practically infinite design space of 2.98×10^{52} with a relatively small deviation of 0.7%
    c. For tall buildings, the design of lower story columns is controlled by the overall stability of the frame rather than the strength of individual columns.

7.3 Future Research

There are many ways to develop new algorithms, and from the metaheuristic point of view, the best way is probably to develop new algorithms by hybridization. That is to say, new algorithms are often based on the right combination of the existing metaheuristic algorithms. For example, combining a trajectory type simulated annealing with multiple agents; the parallel simulated annealing optimization (PSO) can be developed. In the context of ABC algorithms, the combination of ABC with PSO. As in the case of any efficient metaheuristic algorithms, the most difficult thing is probably to find the right or optimal balance between diversity and intensity of the found solutions; here the most challenging task in developing new hybrid algorithms is probably to find the right combination of which feature/components of existing algorithms.

Another great opportunity for future research is to extend the research scope to include:

1) Nonlinear behavior of reinforced concrete frames
2) Larger design variables such as the number of stirrup legs in a beam
3) Including intermediate and special moment resisting frames.
4) Comparing optimization results based on various codes of practice
5) Comparing optimization results based on various optimization techniques
6) Including foundation design and costs into consideration
7) Including seismic load calculation into consideration instead of assuming it.

Nevertheless, the field of optimization of reinforced concrete is full of new ideas that wait to be researched and analyzed.

REFERENCES

1. *ACI 352-02, Recommendations for Design of Beam-Column Connections in Monolithic Reinforced Concrete Structures.* (2002). American Concrete Institute (ACI).

2. *ACI* 318-08m, *Building Code Requirements for Structural Concrete and Commentary.* (2008). American Concrete Institute (ACI).

3. Akay, B., & Karaboga, D. (2009). A comparative study of Artificial Bee Colony Algorithm. *Applied Mathmatics and Computation* (214), 108-132 (2009).

4. Akin, A., & Saka, M. (2010). Optimum Design of Concrete Cantilever Retaining Walls using the Harmony Search Algorithm. *Civil-Comp Press.* Stirlingshire.

5. Akin, A., & Saka, M. P. (2011). *Design optimization of reinforced concrete frames to ACI 318-08, Ph.D. Thesis, Middle East Technical University, Ankara, Turkey.*

6. Arora, J. S. (1989). *Introduction to optimum design.* New York: Mc Graw Hill.

7. Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems.* New York: Oxford University Press.

8. Camp, C., Pezeshk, S., & Hansson, H. (2003). Flexural Design of Reinforced Concrete Frames Using a Genetic Algorithm. *ASCE Journal of Structural Engineering, 129*(1).

9. Coello, C. A., Christiansen, A. D., & Hernandez, F. (1997). A simple Genetic Algorithm for the design of Reinforced Concrete Beams. *Engineering with Computers*(13), 185-196.

10. De Castro, L. N., & Von Zuben, F. J. (1999). *Artificial Immune Systems. Part I. Basic Theory And Applications.* Brazil.

11. Dorigo, M., Maniezzo, V., & Colorni, A. (1992). An investigation of some properties of an ant algorithm. *Parallel Problem Solving from Nature Conference (PPSN 1992).* Brussels.

12. Geem, Z. W., & Lee, K. S. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*(194), 3902-3933.

13. Glover, F. (1977). Heuristic for integer programming using surrogate constraints. *Decision Sci.*(8), 156-166.

14. Hadidi, A., & Kazemzadeh , S. (2010). Structural Optimization using artificial bee colony algorithm. *2nd International Conference on Engineering Optimization.*

15. Hassoun, N., & Al-Manaseer, A. (2008). *Structural Concrete Theory and Design* (4th ed.). John Wiley.

16. Haupt, R. L. (2004). *Practical Genetic Algorithms* (2nd ed.). John Wiley.

17. Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press.

18. Jenkins, W. M. (1991). Towards structural optimization via the genetic algorithm. *Computers and Structures*(40), 1321-1327.

19. Karaboga, D., & Basturk, B. (2008). Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems. *Advances in Soft Computing, Vol (8), Issue (1), 687-697.*

20. Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization,Vol (39), Issue (3), 459-471*.

21. Karboga, D. (2005). *An idea based on honey bee swarm for numerical optimization, Technical Report TR06.* 2005: Erciyes University.

22. Kaveh, A., & Sabzi, O. (2011). A comparative study of two meta-heuristic algorithms for optimum designs of reinforced concrete frames. *Structural Engineering, 9*(3).

23. Kennedy, J., & Eberhart, R. C. (1995). Particle Swarm Optimization. *1995 IEEE International Conference on Neutral Networks*, *4*, pp. 1942-1948.

24. Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by simulated annealing. *Science*(2220), 671-680.

25. Kumar, N. D. (2005). *Classical and Advanced Techniques for Optimization.*

26. Kwak, H., & Kim, J. (2008). An integerated genetic algorithm complemented with direct search for optimum design of RC frames. *Computer-Aided Design*(40), 396-408.

27. McCormac, J. C., & Nelson, J. K. (2006). *Design of Reinforced Concrete* (7th ed.). John Wiley.

28. Metropolis, N., Rosenbluth, A., & Teller, A. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*(21), 1087-1092.

29. Nilsson, H. I. (1976). Reinforced concrete corners and joints subjected to bending moment. *Proceedings ASCE, Journal of the Structural Division,* Vol (102), No. ST6, 16pp.

30. Pincus, M. (1970). A Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Operation Resources*(18), 1225-1228.

31. Rajeev, S., & Krishnamoorthy, C. S. (1998). Genetic Algorithm-Based Methodology for Design Optimization of Reinforced Concrete Frames. *Computer-Aided Civil and Infrastructure Engineering, 13*.

32. Saka, M. P., & Akin, A. (2010). Optimum Detailed Design of Reinforced Concrete Continuous Beams using the Harmony Search Algorithm. *Civil-Comp Press.* Stirlingshire, UK.

33. Shihada, S. (2011). *Design of Earthquake Resistant Structures, Lecture Notes.* Gaza.

34. Shihada, S. (2011). *Design of Reinforced Concrete Structures.* Gaza: Dar Al Manara Book Shop.

35. Stutzle, T. (2010). *Heuristic Optimization Lecture Notes.* University of Bruxelles.

36. Valle, Y., Harley, R. G., & Venayagamoorthy, G. K. (2009). *Comparison of Enhanced-PSO and Classical Optimization Methods: a case study for STATCOM placement.* Missouri University of Science and Technology.

37. Vesterstorm, J., & Riget, J. (n.d.). *Particle Swarm Extensions for improved local, multi-modal, and dynamic search in numerical optimization, IEEE Proceedings of the Congress on Evolutionary Computation* (2002).

38. Wight, J. K., & MacGregor, J. G. (2008). *Reinforced Concrete Mechanics and Design* (5th ed.). Prentice Hall.

39. Zaforteza, I., Yepes, V., & Vidosa, F. (2009). CO2 - optimization of reinforced concrete frames by simulated annealing. *Engineering Structures*(31), 1501-1508.

40. UBC-97, Uniform Building Code, V.2, Structural Engineering Design Provisions, 1997 Edition, International Council of Building Officials, Whittier, CA, 1997.

APPENDIX A : MAIN ABC ALGORITHM CODE

```
%Variable Initializer----------------------------------
NCG = ceil(NStory/NSpCG)*(NBay+1); %Number of Column Groups
NBG = ceil(NStory/NSpBG);      %Number of Beam Groups.
% Preparation of Bays
Spans = ones(1,NBay)*BayLength;
% Preparing Node Coordinates:
NNodes = (NBay+1)*(NStory+1);
Story12 = 1;
Bay12 = 1;
coords = zeros(NNodes,2);
for i = 1:NNodes
    % Coordinates
    x = (Bay12-1)*BayLength;
    y = (Story12-1)*StoryHeight;
    coords(i,:) = [x,y];
    if Story12 == NStory+1
        Bay12 = Bay12+1;
        Story12 = 1;
    else
        Story12 = Story12+1;
    end
end
%Preparing Connectivity Matrix
NColumns = (NBay+1)*NStory;
NBeams = NStory*NBay;
NMember = NColumns+NBeams;
nconnect = zeros(NMember,2);
current_node = 1;
next_node = 2;
for j = 1:NColumns
    nconnect(j,:) = [current_node,next_node];
    if rem(next_node,NStory+1)==0
        current_node = current_node+2;
        next_node = next_node+2;
    else
        current_node = current_node+1;
        next_node = next_node+1;
    end
end
startingbeam = NColumns;
currentstory = 1;
currentbay = 1;
for j = 1:NBeams
    location = startingbeam +j;
    firstnode = (NStory+1)*(currentbay-1)+currentstory+1;
    finalnode = (NStory+1)*currentbay + currentstory+1;
    nconnect(location,:) = [firstnode,finalnode];
    if currentbay == NBay
        currentbay = 1;
        currentstory = currentstory +1;
    else
        currentbay = currentbay+1;
    end
end
%Preparing Joint Force Table
load = zeros(NStory+1,4);
firstcol = (1:NStory+1)';
secondcol = data';
load(:,1) = firstcol;
```

```
load(2:NStory+1,2) = secondcol;
%Prepare Supports
support = zeros(NBay+1,7);
restraint = ones(NBay+1,3);
if pinswitch ==1
    restraint(:,3) = restraint(:,3)*0;
end
support(:,2:4) = restraint;
nodenumbers = [1:NStory+1:NNodes]';
support(:,1) = nodenumbers;
%Member Uniform Loads
BeamNumbers = (startingbeam+1:1:NMember)';
nele = NMember;
emod(1:nele)=210*10^6;
area(1:nele)=0.5*0.5;
Inert(1:nele) = 1/12*0.5*0.5^3;
% Assemble all Variables in Vectors
LowerBounds = [bl,hl,bfnbl,bfdbl,bpnbl,bpdbl,ufnbl,ufdbl,upnbl,updbl,c1l,
c2l,nbl,dbl,dsl];
UpperBounds = bu,hu,bfnbu,bfdbu,bpnbu,bpdbu,ufnbu,ufdbu,upnbu,updbu,c1u,
c2u,nbu,dbu,dsu];
StepSize = [bs,hs,bfnbs,bfdbs,bpnbs,bpdbs,ufnbs,ufdbs,upnbs,updbs,c1s,
c2s,nbs,dbs,dss];
STRAnal.coord = coords;
STRAnal.nconnect = nconnect;
STRAnal.load = load;
STRAnal.support = support;
%STRAnal.memberload = memberload;
STRAnal.emod = emod;
STRAnal.area = area;
STRAnal.Inert = Inert;
STRAnal.DL = DL;
STRAnal.LL = LL;
STRAnal.Case = Case;
% runABC
tic
[~,x1,x2,x3] =
SolutionGenerator(1,LowerBounds,UpperBounds,StepSize,NCG,NBG,NBay);
lb = x1;
ub = x2;
StepSizeDB = x3;
maxCycle = MaxCycle;
FoodNumber=NP/2;
numberParam2Change = Params2Change;
%Foods [FoodNumber][D]; /*Foods is the population of food sources. Each row
of Foods matrix is a vector holding D parameters to be optimized. The
number of rows of Foods matrix equals to the FoodNumber*/
%ObjVal[FoodNumber];  /*f is a vector holding objective function values
associated with food sources */
%Fitness[FoodNumber]; /*fitness is a vector holding fitness (quality)
values associated with food sources*/
%trial[FoodNumber]; /*trial is a vector holding trial numbers through which
solutions can not be improved*/
%prob[FoodNumber]; /*prob is a vector holding probabilities of food sources
(solutions) to be chosen*/
%solution [D]; /*New solution (neighbour) produced by
v_{ij}=x_{ij}+\phi_{ij}*(x_{kj}-x_{ij}) j is a randomly chosen parameter
and k is a randomlu chosen solution different from i*/
%ObjValSol; /*Objective function value of new solution*/
%FitnessSol; /*Fitness value of new solution*/
```

```
%neighbour, param2change; /*param2change corrresponds to j, neighbour
corresponds to k in equation v_{ij}=x_{ij}+\phi_{ij}*(x_{kj}-x_{ij})*/
%GlobalMin; /*Optimum solution obtained by ABC algorithm*/
%GlobalParams[D]; /*Parameters of the optimum solution*/
%GlobalMins[runtime]; /*GlobalMins holds the GlobalMin of each run in
multiple runs*/
GlobalMins=zeros(1,runtime);
[smthn,~,~,~] = SolutionGenerator(1,...
          LowerBounds,UpperBounds,StepSize,NCG,NBG,NBay);
nn = size(smthn,2);
GlobalParamsDB = zeros(runtime,nn);
y_axis = zeros(runtime,maxCycle);
x_axis = 1:1:maxCycle;
LastImprovementDB = zeros(1,runtime);
for r=1:runtime
    set(handles.RuntimeStatus,'String',r)
    ObjVal = zeros(1,FoodNumber);
    Penalty = ObjVal;
    Initial = ObjVal;
    % /*All food sources are initialized */
    %/*Variables are initialized in the range [lb,ub]. If each parameter
has different range, use arrays lb[j], ub[j] instead of lb and ub */
    [Foods,~,~,~] = SolutionGenerator(FoodNumber,...
        LowerBounds,UpperBounds,StepSize,NCG,NBG,NBay);
    if Symmetry == 1
        for i = size(Foods,1)
            tempFood = Foods(i,:);
            SymmetrifiedFood =
SolutionSymmetrifier(tempFood,NCG,NBG,NStory,NBay);
            Foods(i,:) = SymmetrifiedFood;
        end
    end
    for i = 1:FoodNumber
        [Initial(i),Penalty(i)] =
ObjectiveFunction(Foods(i,:),fc,fy,Cs,Cc...

,Cf,formworkswitch,NBay,NStory,NSpCG,NSpBG,STRAnal,Spans,StoryHeight,
selfweightswitch,jointswitch,shearswitch,slendernesslimit);
        ObjVal(i) = Penalization(Initial(i),Penalty(i));
    end
   %reset trial counters
    trial=zeros(1,FoodNumber);
    %/*The best food source is memorized*/
    BestInd=find(ObjVal==min(ObjVal));
    BestInd=BestInd(end);
    GlobalMin=ObjVal(BestInd);
    GlobalParams=Foods(BestInd,:);
    D = size(Foods,2); % Number of Decision Variables
    iter=1;
    while ((iter <= maxCycle)),
        set(handles.IterationStatus,'String',iter)
        pause(0.01)
        %%%%%%%%% EMPLOYED BEE PHASE %%%%%%%%%%%%%%%%%%%%%%%
        for i=1:(FoodNumber)

            %/*The parameter to be changed is determined randomly*/
            Param2Change=fix(rand(1,numberParam2Change)*D)+1;

            %/*A randomly chosen solution is used in producing a mutant
solution of the solution i*/
            neighbour=fix(rand*(FoodNumber))+1;
```

```
            %/*Randomly selected solution must be different from the
solution i*/
            while(neighbour==i)
                neighbour=fix(rand*(FoodNumber))+1;
            end;
                 sol=Foods(i,:);
                 %oldsol = sol
                 %  /*v_{ij}=x_{ij}+\phi_{ij}*(x_{kj}-x_{ij}) */
            sol(Param2Change)=Foods(i,Param2Change)+(Foods(i,Param2Change)-
Foods(neighbour,Param2Change))*(rand-0.5)*6;
            %bb = input('!!')
            %  /*if generated parameter value is out of boundaries, it is
shifted onto the boundaries*/
            ind=find(sol<lb);
            sol(ind)=lb(ind);
            ind=find(sol>ub);
            sol(ind)=ub(ind);
             % Round Each Variable to its correspondend step
            for j = 1:length(sol)
                first_part = fix(sol(j)/StepSizeDB(j))*StepSizeDB(j);
                second_part =round((sol(j)-
fix(sol(j)/StepSizeDB(j))*StepSizeDB(j))/StepSizeDB(j))*StepSizeDB(j);
                sol(j) = first_part+second_part;
            end
            if Symmetry == 1
                sol = SolutionSymmetrifier(sol,NCG,NBG,NStory,NBay);
            end
            %evaluate new solution
                 [Initial,Penalty] =  ObjectiveFunction(sol,fc,fy,Cs,Cc...
,Cf,formworkswitch,NBay,NStory,NSpCG,NSpBG,STRAnal,Spans,StoryHeight,
selfweightswitch,jointswitch,shearswitch,slendernesslimit);
            ObjValSol = Penalization(Initial,Penalty);

            % /*a greedy selection is applied between the current solution
i and its mutant*/
            if (ObjValSol<ObjVal(i)) %/*If the mutant solution is better
than the current solution i, replace the solution with the mutant and reset
the trial counter of solution i*/
                Foods(i,:)=sol;
                ObjVal(i)=ObjValSol;
                trial(i)=0;
            else
                trial(i)=trial(i)+1; %/*if the solution i can not be
improved, increase its trial counter*/
            end;
        end;
%%%%%%%%% CalculateProbabilities%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%/* A food source is chosen with the probability which is proportioal to
its quality*/
%/*Different schemes can be used to calculate the probability values*/
%/*For example prob(i)=fitness(i)/sum(fitness)*/
%/*or in a way used in the metot below
prob(i)=a*fitness(i)/max(fitness)+b*/
%/*probability values are calculated by using fitness values and
%normalized by dividing maximum fitness value*
        prob=(0.9.*min(ObjVal)./ObjVal)+0.1; %The Better the Higher

%%%%%%%%%%%%%%%% ONLOOKER BEE PHASE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        i=1;
        t=0;
        while(t<FoodNumber)
```

```
            if(rand<prob(i))
                t=t+1;
                %/*The parameter to be changed is determined randomly*/
                Param2Change=fix(rand(1,numberParam2Change)*D)+1;
                %/*A randomly chosen solution is used in producing a mutant
solution of the solution i*/
                neighbour=fix(rand*(FoodNumber))+1;
%/*Randomly selected solution must be different from the solution i*/
                while(neighbour==i)
                    neighbour=fix(rand*(FoodNumber))+1;
                end;
                sol=Foods(i,:);
                %  /*v_{ij}=x_{ij}+\phi_{ij}*(x_{kj}-x_{ij}) */
                  sol(Param2Change)=Foods(i,Param2Change)+(Foods(i,Param2Ch
                  ange)-Foods(neighbour,Param2Change))*(rand-0.5)*6;
                  %  /*if generated parameter value is out of boundaries,
                  it is shifted onto the boundaries*/
                ind=find(sol<lb);
                sol(ind)=lb(ind);
                ind=find(sol>ub);
                sol(ind)=ub(ind);
                % Round Each Variable to its correspondend step
                for j = 1:length(sol)
                    first_part = fix(sol(j)/StepSizeDB(j))*StepSizeDB(j);
                    second_part =round((sol(j)-
fix(sol(j)/StepSizeDB(j))*StepSizeDB(j))/StepSizeDB(j))*StepSizeDB(j);
                    sol(j) = first_part+second_part;
                end
                if Symmetry == 1
                    sol = SolutionSymmetrifier(sol,NCG,NBG,NStory,NBay);
                end
                %evaluate new solution
[Initial,Penalty] =  ObjectiveFunction(sol,fc,fy,Cs,Cc...

,Cf,formworkswitch,NBay,NStory,NSpCG,NSpBG,STRAnal,Spans,StoryHeight,
selfweightswitch,jointswitch,shearswitch,slendernesslimit);
                ObjValSol = Penalization(Initial,Penalty);              %
/*a greedy selection is applied between the current solution i and its
mutant*
               if (ObjValSol<ObjVal(i)) %/*If the mutant solution is better
than the current solution i, replace the solution with the mutant and reset
the trial counter of solution i*/
                    Foods(i,:)=sol;
                    ObjVal(i)=ObjValSol;
                    trial(i)=0;
                else
                    trial(i)=trial(i)+1; %/*if the solution i can not
beimproved, increase its trial counter*/
                end;
            end
            i=i+1;
             if (i==(FoodNumber)+1)
                i=1;
            end;
        end;
       %/*The best food source is memorized*/
        ind=find(ObjVal==min(ObjVal));
        ind=ind(length(ind));
        if (ObjVal(ind)<GlobalMin)
            GlobalMin=ObjVal(ind);
            GlobalParams=Foods(ind,:);
```

```
            [Initial,Penalty] =  ObjectiveFunction(GlobalParams,fc,fy,Cs...

            ,Cc,Cf,formworkswitch,NBay,NStory,NSpCG,NSpBG,STRAnal,Spans,Sto
            ryHeight,
            selfweightswitch,jointswitch,shearswitch,slendernesslimit);
            set(handles.GlobalOptimaStatus,'String',GlobalMin)
            set(handles.GlobalOptimaICStatus,'String',Initial)
            set(handles.GlobalOptimaPenaltyStatus,'String',Penalty)
            pause(0.01)
        end;
        %%%%%%%%%%%% SCOUT BEE PHASE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%/*determine the food sources whose trial counter exceeds the "limit"
value.%In Basic ABC, only one scout is allowed to occur in each cycle*/
            ind=find(trial==max(trial));
        ind=ind(end);
        if (trial(ind)>limit)
            trial(ind)=0;
            [sol,~,~,~] = SolutionGenerator(1,...
                LowerBounds,UpperBounds,StepSize,NCG,NBG,NBay);
            if Symmetry == 1
                sol = SolutionSymmetrifier(sol,NCG,NBG,NStory,NBay);
            end
            [Initial,Penalty] =  ObjectiveFunction(sol,fc,fy,Cs,Cc...

            ,Cf,formworkswitch,NBay,NStory,NSpCG,NSpBG,STRAnal,Spans,StoryH
            eight,selfweightswitch,jointswitch,shearswitch,slendernesslimit
            );
       ObjValSol = Penalization(Initial,Penalty);
            ObjVal(ind)=ObjValSol;
        end
  %GlobalMin
        y_axis(r,iter) = GlobalMin;
        if iter~=1
            current = y_axis(r,iter);
            previous = y_axis(r,iter-1);
            if current~=previous
                set(handles.LastChangeStatus,'String',iter)
            end
        end
        iter=iter+1;
    end % End of ABC
    LastImprovementDB(r) = str2num(get(handles.LastChangeStatus,'String'));
    GlobalParamsDB(r,:) = GlobalParams;
    GlobalMins(r)=GlobalMin;
end %end of runs
x =toc;
```

APPENDIX B: OPTIMIZATION MODEL'S GRAPHICAL USER INTERFACE (USER MANUAL)

This appendix presents the main components of the graphical user interface developed to input various fixed parameters into the optimization model and demonstrate how to perform optimizations using this user interface.

1) Open MatLab and run the "projectGUI.m" file. This will open up the main graphical user interface, shown in Figure (A. 1), that starts with some general information about the program. The two buttons at the bottom of this GUI interface are used to navigate between various pages that require user input. To proceed, the (>) button has to be pressed.

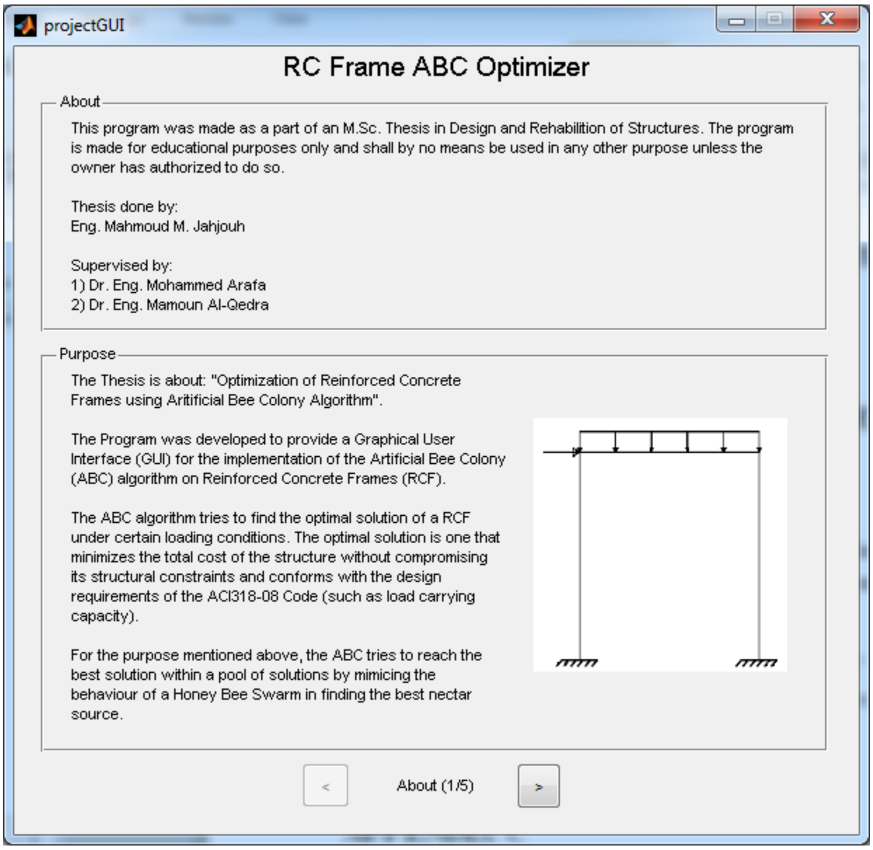


Figure A. 1: First page in GUI

2) The second page, shown in Figure (A. 2), is used to define the shape of the structure as well as the loads that are applied on the structure. Figures are included to further aid the user in understanding the terminology used in the GUI. Once all inputs are set, one has to proceed to the next page for further inputs.

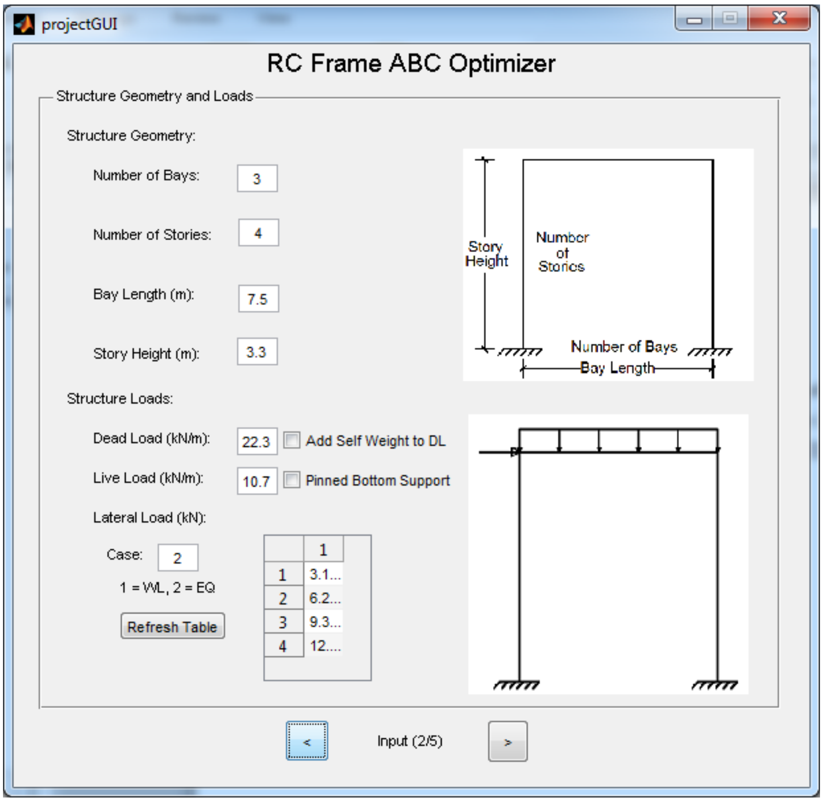


Figure A. 2: Structural and Load Inputs

3) The third page, shown in Figure (A. 3), is used to define the structural member groups, material characteristics and cost inputs.

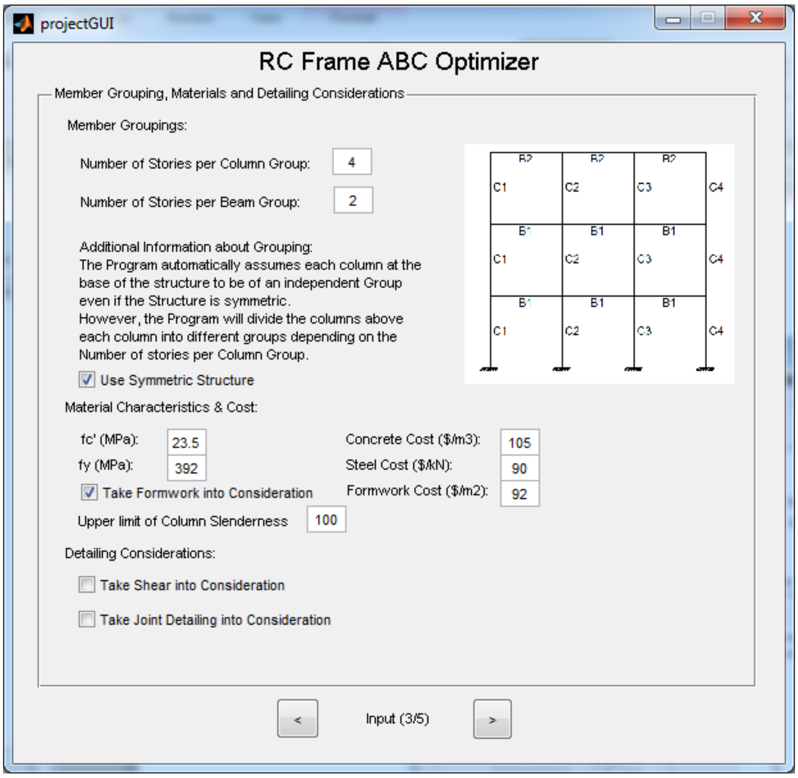


Figure A. 3: Groups, Materials and Cost inputs

4) The fourth page, shown in Figure (A. 4), is used to input the upper limits, lower limits and step sizes for the various design parameters used in the design of reinforced concrete frames.

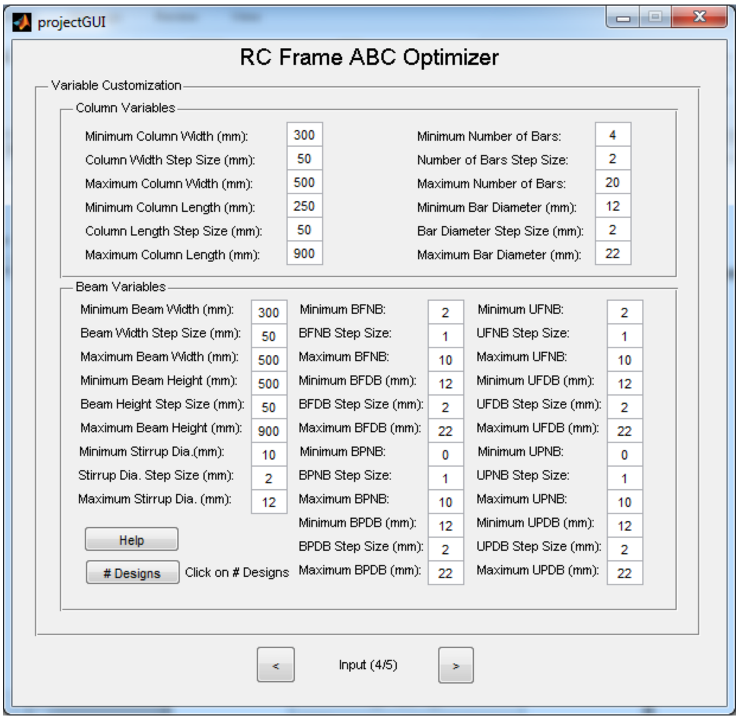


Figure A. 4: Upper limits, lower limits and step size input page

5) The final page, Figure (A. 5), is used to input the ABC algorithm control parameters, and provides an option to save the inputs, solutions and to load them back to encourage software reusability. Furthermore, this page is responsible of starting the optimization process and gives “real time” updates of the current situation such as the current global optima, constraint violation, elapsed time, iteration, run, as well as the last iteration at which the global optimum has improved.
6) Once the optimization is complete, the GUI will automatically save the best solution into the designated path, and shows the convergence history for the optimization problem.
7) Another useful option is to open AutoCAD and have the program automatically draw the reinforced concrete frame. This option is still under evaluation.

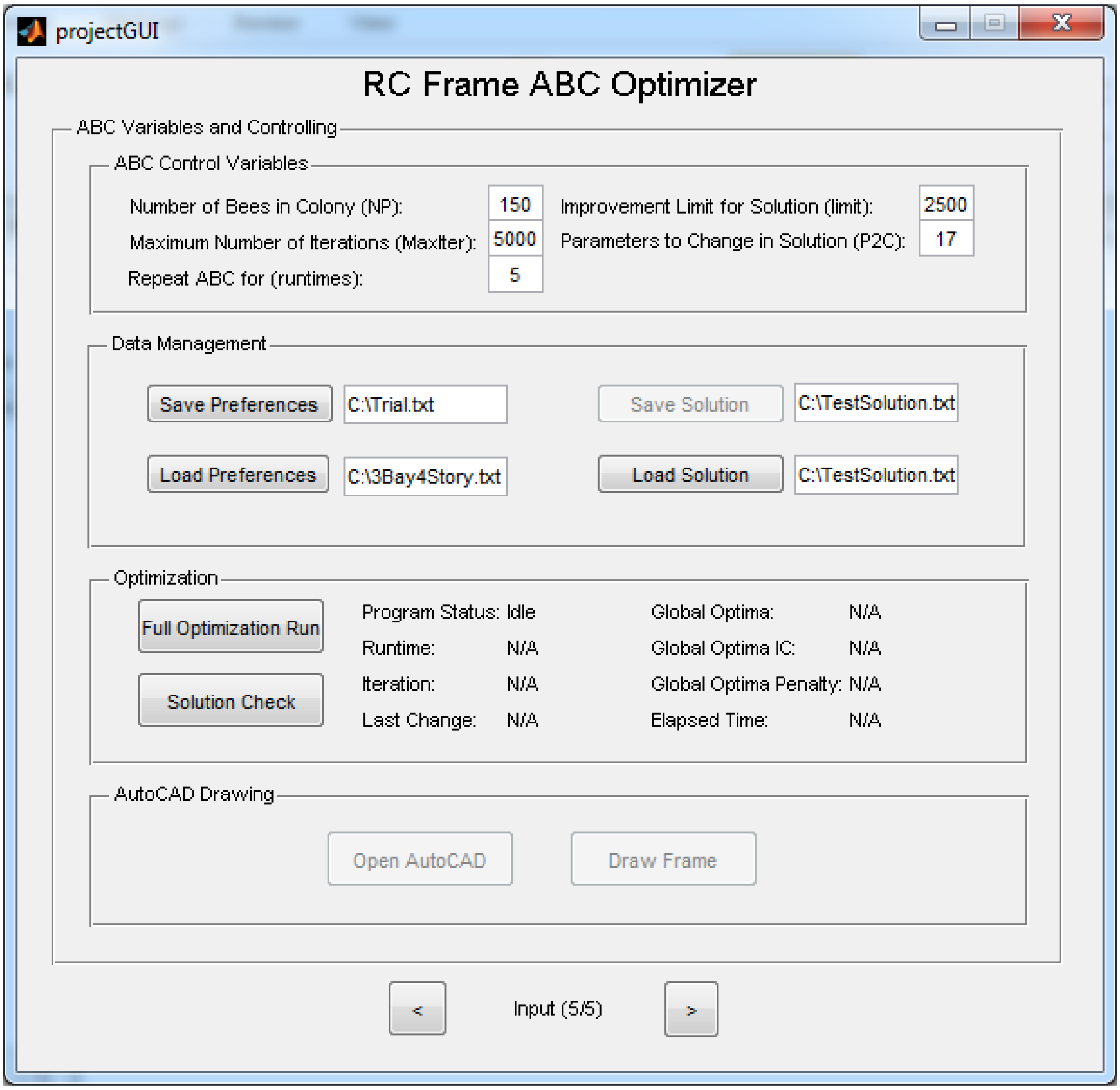


Figure A. 5: Last page in GUI

APPENDIX C: DETAILED RESULTS FOR ONE BAY ONE STORY FRAME

1) Optimization Results using various values for *VCP*:

Table A. 1: Optimization Results using various values for *VCP*

Test Program	*VCP*	Test Run	Last Improvement	Best Cost ($)
1	10%	1	1563	3539.6
		2	1696	3621.6
		3	1786	3539.6
		4	788	3536.9
		5	1754	3537.7
		6	1961	3539.6
		7	1823	3536.5
		8	1906	3539.6
		9	1687	3536.9
		10	1250	3540.8
2	20%	1	1280	3539.6
		2	1090	3536.5
		3	775	3536.5
		4	1625	3536.5
		5	1969	3536.5
		6	1614	3536.5
		7	1572	3536.5
		8	814	3536.5
		9	1909	3536.5
		10	906	3536.5
3	30%	1	1271	3536.5
		2	1450	3536.5
		3	1620	3536.5
		4	850	3539.6
		5	1332	3536.5
		6	1555	3536.5
		7	1203	3536.5
		8	980	3536.5
		9	1233	3536.5
		10	887	3536.5

Table A.1: cont'd

Test Program	*VCP*	Test Run	Last Improvement	Best Cost ($)
4	40%	1	924	3536.5
		2	752	3536.5
		3	1312	3536.5
		4	1314	3536.5
		5	1017	3536.5
		6	863	3536.5
		7	1157	3536.5
		8	1089	3539.6
		9	1206	3536.5
		10	701	3536.5
5	50%	1	807	3539.6
		2	535	3536.5
		3	975	3539.6
		4	670	3539.6
		5	918	3539.6
		6	1246	3536.5
		7	952	3539.6
		8	635	3536.5
		9	736	3536.5
		10	919	3539.6
6	60%	1	1660	3583.3
		2	474	3841.7
		3	371	3536.5
		4	448	3539.6
		5	549	3536.5
		6	732	3539.6
		7	827	3536.5
		8	689	3536.5
		9	558	3945.5
		10	635	3841.7

Table A.1: cont'd

Test Program	*VCP*	Test Run	Last Improvement	Best Cost ($)
7	70%	1	980	3539.6
		2	737	3763.4
		3	507	3536.5
		4	623	3583.3
		5	561	3536.5
		6	343	3815.8
		7	336	3841.7
		8	254	3841.7
		9	280	4021.4
		10	615	3536.5

2) Optimization Results using various values for N_P

Table A. 2: Optimization Results using various values for N_P

Test Program	N_P	Test Run	Last Improvement	Best Cost ($)
1	10	1	121	4175.9
		2	273	4595.8
		3	178	4169.8
		4	388	4187.5
		5	196	4059.1
		6	113	6385.6
		7	62	4414
		8	68	4890.6
		9	401	4294.9
		10	211	3856.6
2	20	1	1013	3845.6
		2	820	4136
		3	800	3998.4
		4	439	3600.9
		5	455	3777.7
		6	562	3983.9
		7	344	3955.7
		8	533	4660.7
		9	778	3579.2
		10	361	3817.9

Table A.2: cont'd

Test Program	N_P	Test Run	Last Improvement	Best Cost ($)
3	30	1	1312	3536.5
		2	732	3583.3
		3	713	3540.8
		4	442	3565.1
		5	625	3539.6
		6	888	3539.6
		7	910	3536.5
		8	420	3567.4
		9	1288	3606.8
		10	1117	3537.7
4	40	1	1494	3536.5
		2	1349	3536.5
		3	773	3539.6
		4	499	3723.5
		5	1163	3536.5
		6	1170	3536.5
		7	587	3536.5
		8	416	3761.3
		9	599	3764.6
		10	1053	3539.6
5	50	1	1231	3539.6
		2	870	3676.5
		3	1376	3539.6
		4	804	3536.5
		5	799	3536.5
		6	690	3536.5
		7	688	3536.5
		8	546	3563.1
		9	666	3539.6
		10	898	3834.1

Table A.2: cont'd

Test Program	N_P	Test Run	Last Improvement	Best Cost ($)
6	60	1	619	3536.5
		2	648	3594.9
		3	670	3539.6
		4	1343	3536.5
		5	897	3536.5
		6	1142	3536.5
		7	948	3536.5
		8	1078	3536.5
		9	1034	3536.5
		10	1370	3536.5
7	70	1	924	3536.5
		2	752	3536.5
		3	1312	3536.5
		4	1314	3536.5
		5	1017	3536.5
		6	863	3536.5
		7	1157	3536.5
		8	1089	3539.6
		9	1206	3536.5
		10	701	3536.5
8	80	1	1215	3539.6
		2	1235	3536.5
		3	1041	3537
		4	785	3536.5
		5	468	3536.5
		6	654	3536.5
		7	717	3537.7
		8	912	3536.5
		9	772	3536.5
		10	1038	3539.6

Table A.2: cont'd

Test Program	N_P	Test Run	Last Improvement	Best Cost ($)
9	90	1	893	3539.6
		2	780	3536.5
		3	751	3537.7
		4	713	3536.5
		5	954	3536.5
		6	831	3536.5
		7	759	3536.5
		8	814	3536.5
		9	964	3536.5
		10	764	3536.5
10	100	1	862	3536.5
		2	598	3536.5
		3	740	3536.5
		4	883	3536.5
		5	1225	3536.5
		6	776	3536.5
		7	627	3536.5
		8	650	3536.5
		9	544	3536.5
		10	641	3536.5
11	110	1	762	3536.5
		2	598	3536.5
		3	740	3536.5
		4	883	3536.5
		5	925	3536.5
		6	776	3536.5
		7	627	3536.5
		8	650	3536.5
		9	544	3536.5
		10	641	3536.5