An Object-Oriented Evolutionary Algorithm for Automated Advanced Analysis Based Design

Daniel Schinler
Marquette University
Dept. of Civil & Environmental Engineering
Milwaukee, WI 53233

Christopher M. Foley
Marquette University
Dept. of Civil & Environmental Engineering
Milwaukee, WI 53233

Abstract

Genetic and evolutionary algorithms have long been used for optimized design of building structures. Optimized design using these techniques has most often utilized binary strings to represent individuals. Object oriented programming (OOP) affords convenient encapsulation of building components, which in turn allows alternate methods for representing design variables within EA's. Advanced analysis based optimization combines inelastic analysis and constraints in such a manner that design specifications and codes are not needed. Therefore, advanced analysis-based design has many of the elements of performance-based design. The present paper outlines and evaluates an automated design procedure that uses an object-oriented evolutionary algorithm with advanced analysis to design steel frameworks.

1 INTRODUCTION

Binary string representations of design variables has been the de-facto standard means with which to implement design optimization using genetic and evolutionary algorithms. Variations on a theme have been developed (Parmee 1995), but there has been little research related to new mechanisms by which structural engineers can represent design variables.

Object-oriented programming (OOP) has created a very convenient way to accomplish encapsulation in computer programs. The encapsulation (packaging) offered by OOP has resulted in structural engineers formulating new ways to represent building structures on the computer (Rivard and Fenves 2000). A building structure can be visualized as a hierarchy of objects as shown in Figure 1. Encapsulating building components as objects creates a mechanism by which system optimization can occur. Consider two individuals present in a generational snapshot of the evolution (Figure 2). Individual 1 (the control individual) is a hybrid structural steel-concrete framework where a moment resisting frame interacts with concrete shear walls to resist lateral loading. Individual 2 contains an eccentrically braced frame (EBF) bay.

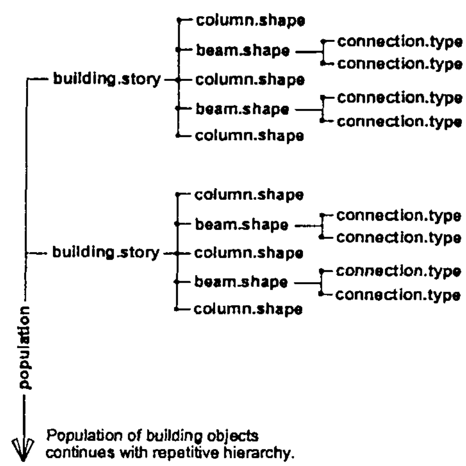

Figure 1: Building Object Hierarchy

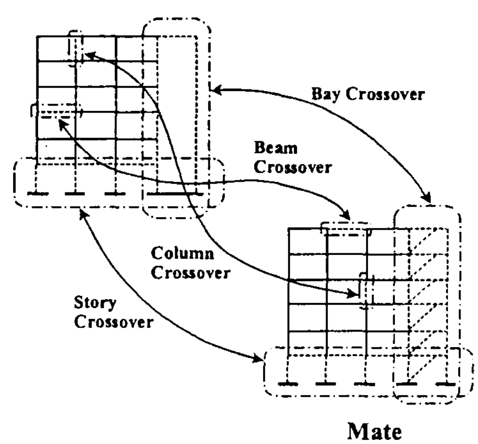

Figure 2: Object Crossover in Buildings

If the binary string representation was used to accomplish crossover, the allele describing a shear wall would most likely be drastically different than the allele describing an EBF. As a result, crossover of these two elements might not make sense.

However, to the structural engineer, swapping shear wall and EBF objects between individuals may result in more efficient hybrid frameworks.

2 OBJECT-ORIENTED EVOLUTIONARY ALGORITHM

An OO-EA differs from the binary genetic algorithm in the manner which crossover and mutation affects the individuals in the population (Figure 3).

	x_1	x_2
INDIVIDUAL A	1101	0011
INDIVIDUAL B	0010	1101
	↑ 1	↑ 2
One Point at 1	1110	1101
	0001	0011
One Point at 2	1101	1101
	0010	0011
Uniform With Mask	1011	1111
0110 1101	0100	0001

Figure 3: Crossover Operation Comparison

Assume that the four-bit binary strings represent wide-flange shapes. Traditional one point crossover at point 1 creates a new shape in the offspring which can be considered as mutation at the design variable level. Variables x_2 remain intact (but are exchanged). If one-point crossover at point 2 occurs, design variables are exchanged. Uniform cross-over creates new genetic material for both design variables.

Object crossover acts much like one-point crossover at point 2 in Figure 3. It does not create new genetic material in the offspring, but merely exchanges it. As a result, the crossover and mutation operations are completely divorced from one another.

A quick example can be formulated to illustrate the behavior of an object oriented EA. Consider the optimization problem outlined below:

$$\textit{Minimize:} \quad y(\mathbf{X}) = \frac{3}{4}x_1^{\,2} + x_1x_2 + \frac{5}{4}x_2^{\,2}$$

$$\textit{Subject To:} \quad \begin{aligned} x_1 + x_2 &\geq 8 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

The optimal solution to this problem has been computed as $\mathbf{X} = \lfloor 6 \quad 2 \rfloor$ by Jenkins (1991).

An object-oriented evolutionary algorithm was written to solve the problem. Two design variable objects were used in a very small hierarchy (Figure 4).

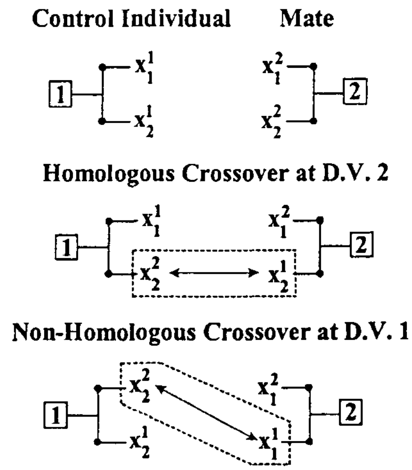

Figure 4: Simple Object Crossover for Two Variable Optimization Problem.

Two types of design variable crossover were implemented (Figure 4). Homologous crossover can be seen to merely swap genes (*ie.* design variables), while non-homologous crossover moves the genes along the chromosome. Mutation is a random change in value within a predefined range.

Research has been undertaken to elucidate the disruptive and beneficial nature of binary crossover and mutation in the evolutionary process (Wu, *et.al.* 1997). The present object formulation has the possibility to foster the creation and retention of beneficial building blocks. However, the object representation requires crossover and mutation be studied to ensure exploration of the design space.

Numerical simulation was undertaken to empirically explore the object representation and its ability to reliably find the solution that minimizes the previously discussed function. Design variables were restricted to integer values, $x_i \in \{1 \to 15\}$. A general crossover probability, p_c was defined as well as subsequent probability of non-homologous crossover, p_{nh}. The probability of design variable mutation, p_m, was also defined,. Linear scaling of the fitness values was used and fitness proportional (roulette wheel) selection was employed (Jenkins 1991). Success of the object-oriented evolutionary algorithm is defined as finding the optimal solution.

A small emprical study was undertaken to evaluate the effect of population size on the success of the evolutionary algorithm proposed. The results of this study are given in Table 1.

Table 1: Effect of Population Size on Success of OO Evolutionary Algorithm: $p_c = 0.60$, $p_{nh} = 0.0$.

Population Size	Mutation Rate	Success Rate (%)
20	0.25	53
30	0.25	73
40	0.25	93
	0.30	93

It was also decided to evaluate the effect of mutation rate on the smallest population. This effect on the OO-EA's success is given in Table 2.

Table 2: Effect of Mutation Rate on Success of OO Evolutionary Algorithm: $p_c = 0.60$, $p_{nh} = 0.0$, $N_{indiv} = 20$.

Mutation Rate	Success Rate (%)
0.25	53
0.40	53
0.60	80
0.80	93
1.00	87

Voss and Foley (1999) studied the effect of non-homologous crossover on a simple cantilever optimization problem. The genetic representation (although hierarchical) in this former study did not utilize objects and therefore, it was decided to study the effect of non-homologous crossover on the OO-EA (Table 3).

Table 3: Effect of Population Size and Mutation Rate on Success of OO Evolutionary Algorithm: $p_c = 0.60$.

Population Size	Non-Homologous Crossover Rate	Success Rate (%)
20, $p_m = 0.25$	0.20	40
	0.40	40
	0.60	53
	0.80	67
	1.00	67
40, $p_m = 0.30$	0.20	93
	0.40	80
	0.60	93
	0.80	87
	1.00	60

The success rate of the OO-EA appears to be linked to the population size. For small populations, the mutation rate should remain high so that sufficient new genetic material can be introduced in the population during the evolution. If a small population is used, non-homologous crossover can improve the evolutionary search. However, the improvement is not as appreciable as mutation. When population sizes are sufficiently large (40 individuals in this case), the non-homologous crossover has less impact on the success rate.

Crossover operations are important to the OO-EA as with binary genetic algorithms. However, one should recognize that the non-homologous operator is important when more design variables are present and/or the hierarchical representation is "deep" (Voss and Foley 1999). This ensures that the crossover operations are able to move the genetic material around and explore building block formation.

3 ADVANCED ANALYSIS BASED DESIGN OF FR AND PR FRAMES

Advanced analysis of steel frames is related to the plastic design methodologies used for rigid frame steel buildings. The goal of advanced analysis is to create a design basis capable of omitting specification equations (LRFD 1999). However, a requirement of advanced analysis based design is that the analysis employed should be capable of considering all critical aspects of steel member behavior used to develop the specification design equations (SSRC 1988). Excellent reviews related to the basic assumptions and requirements of advanced analysis are available (Bridge, *et.al.* 1998; White 1993).

An advanced analysis is capable of addressing individual member strengths in light of the redundancy present in the framework. Therefore, performance-based design criteria are easily incorporated into optimized design algorithms that employ advanced analysis.

3.1 OBJECTIVE FUNCTION

The objective function is based on modified frame weight (Xu, *et.al.* 1995),

$$W = \sum_{k=1}^{N_{col}} L_k A_k \rho_k + \sum_{m=1}^{N_{beam}} L_m A_m \rho_m \left[\left(\sum_{n=1}^{N_{conn}} \zeta_n \right) - 1 \right] \quad (1)$$

where: L is the member length; A is the cross-sectional area; and ρ is the material density. The modifying factor, ζ, accounts for the connection at the beam ends.

Partially and fully restrained connection variations were defined using non-dimensional curves (Bjorhovde, *et.al.* 1990). These curves were established in such a manner that a full range of connection strength and stiffness were present. Fully restrained connections were denoted as C1 while flexible connections were denoted C5. Three connections existed between these limits.

3.2 CONSTRAINTS

Advanced analysis based design allows low-level behavioral constraints to be established. This in-turn allows performance constraints to be easily incorporated into optimization algorithms. Constraints must be established at both service and ultimate load levels.

Service load level constraints used in the advanced analysis based design optimization were:

1. applied load ratio for load combinations,
2. connection rotations,
3. lateral (inter-story) drift,
4. vertical deflection of beams,
5. cross-section plastification.

Ultimate (strength) load level constraints were:

1. applied load ratio for load combinations,
2. connection rotations,
3. plastic hinge curvature,
4. out-of-plane compression buckling,
5. unbraced length for lateral torsional buckling.

An additional constraint, designer preference, was included. This constraint was established to ensure the column below has a nominal depth and weight larger than the column above.

4 EVOLUTIONARY ALGORITHM

An evolutionary algorithm was used to automate the design of partially and fully restrained steel frame configurations with fixed topology. The small empirical study discussed previously was used to initially assign mutation and crossover rates.

4.1 FITNESS

The fitness statement used for the evolutionary algorithm is similar to that suggested by Pezeshk, *et.al.* (2000),

$$f = W\prod_{i=1}^{m}\Phi_i \qquad (2)$$

where: W is defined in (1) and Φ_i are total penalties corresponding to the constraints in the problem.

4.2 PENALTIES

Penalties were written in a form suitable for inclusion in equation (2). For example, the plastic hinge rotation penalty at ultimate load levels is written as,

$$\phi_\kappa = \frac{\kappa}{\kappa_{\text{limit}}} \le 1.0 \qquad (3)$$

where the plastic hinge curvature limit is taken from Yura, *et.al.* (1978),

$$\kappa_{\text{limit}} = 4\kappa_y = \frac{8F_y d_b}{E}$$

All penalties including the designer preference penalty were scaled using (Camp, *et.al.* 1998),

$$p_i = 1.0 + k(\phi_i - 1)^\alpha \qquad (4)$$

The exponent, α, was taken to be 1.0 indicating linear scaling. The parameter k was taken to be 5.0. It should be noted that equation (4) is only applicable when $\phi_i > 1.0$.

The total penalty for plastic hinge rotation at ultimate load levels is computed using,

$$\Phi_\kappa = \prod_{r=1}^{N_U}\prod_{m=1}^{N_{\text{members}}}(p_\kappa)_{m,r} \qquad (5)$$

where: N_U is the number of ultimate load cases (3 in this study); N_{members} is the number of members.

Further details regarding the formulation of the penalties used in this study can be found in Schinler (2001).

4.3 SELECTION

Population partitioning (Camp, *et.al.* 1998; Pezeshk, *et.al.* 2000) is used for selecting individuals for mating. This scheme allows the selection pressure to be controlled and has been found to be effective for the building optimization problems studied. Tournaments were then utilized to assign the mating pool.

4.4 CROSSOVER AND MUTATION

Crossover is performed after choosing two individuals from the mating pool constructed using the selection mechanism. Each individual in this new population is then considered a control individual (refer to Figure 2). A mate (not the same individual) is then chosen from the mating pool. Crossover results in one new individual being created.

In the present study, each column, beam, and connection object in the control individual is chosen for possible crossover. If a random number satisfies the crossover probability, p_c, then crossover will take place with the object in the control individual being exchanged with a similar object in the mate. A second random number is called and if it satisfies the non-homologous crossover rate, p_{nh}, non-homologous crossover occurs. If it does not satisfy this rate, homologous crossover is performed.

Mutation is performed at the object level corresponding to buildings, stories, beams, columns, and connections. A mutation rate, p_m, is established for each of these objects. Mutation takes place after the crossover operations.

5 FRAME DESIGNS

The frame chosen for implementation of the object-oriented evolutionary algorithm is the same as the one used by Xu, *et.al.* (1995). Figure 6 provides an illustration of the framework loading and topology. All beams and columns are assumed to be Grade A36 steel. The connections used in the present study follow the non-dimensional connection curves previously described.

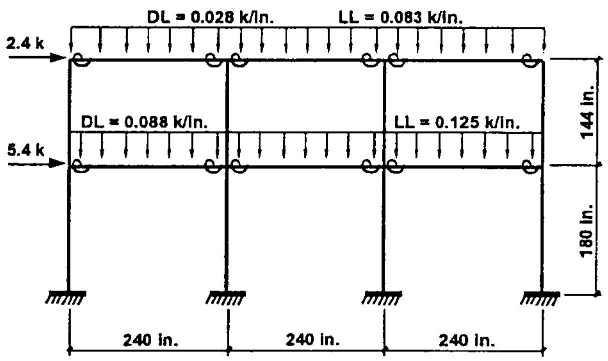

Figure 6: Steel Frame Used in the Present Study.

The interior columns and exterior columns are grouped within each story. Furthermore, the beams in any story are assumed to be the same (including connections). The connections at each end of the beams are required be the same. As a result, the present problem has a total of 6 design variables for the fully restrained (FR) case and 8 design variables for the partially restrained (PR) frame.

The inelastic analysis used to establish an individual's fitness is based on the distributed plasticity model. Zero length connection elements are used on all beams. Details of this inelastic analysis model can be found in Foley and Vinnakota (1999). Other details regarding the OO-EA can be found in Schinler (2001).

The evolutionary algorithm parameters used in the analyses are as follows. The mutation rate was kept constant throughout the evolution. The fixed rate for beams, columns and connections (where applicable) was 30%. Crossover was restricted to columns, beams and connections. The probability of crossover was set at 60% with the subsequent rate of non-homologous crossover set at 50%. The partitioning scheme resulted in the mating pool being developed from the top 40% of the population. Individuals were selected to enter two-at-a-time tournaments from this upper partition 70% of the time.

Results for the two frame designs with comparison to the results obtained by Xu, *et.al.* (1995) are found in Table 6. Connection C4 is one step up from flexible (C5).

Table 6: Frame Weight Comparisons (lbs.)

Frame	Xu, *et.al.* (1995)	Present	
FR	7,031		7,086
PR	6,712	Floor - C4 Roof – C4	6,468

Convergence trajectories of the fittest feasible individual are given in Figures 7 and 8.

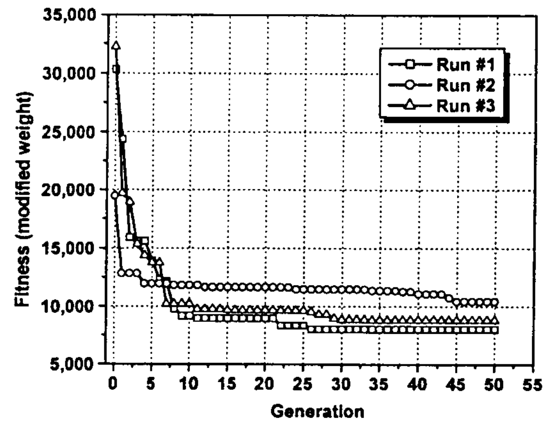

Figure 7: FR Frame Convergence Trajectories for the Fittest Feasible Individual.

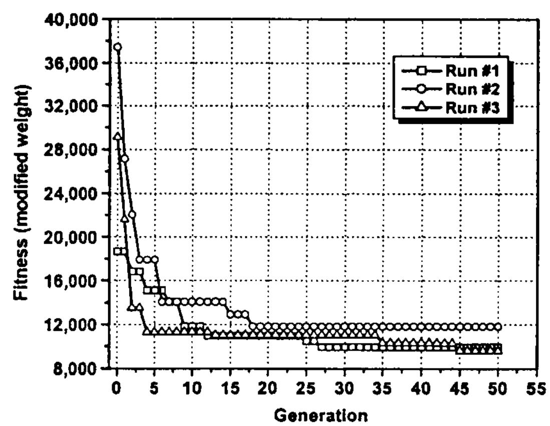

Figure 8: PR Frame Convergence Trajectories for the Fittest Feasible Individual.

The evolutionary algorithm developed exhibits stable convergence characteristics. It should be noted that the fitness illustrated in the figures is the modified weight. This weight includes connection modification factors. As one can see, the same individual does not result for all the runs. Further study of the results during the evolution revealed that the algorithm was making concessions among design variables and constraints and therefore, the same individual might not be expected.

Table 6 illustrates that the present algorithm and the optimization problem formulated gives results consistent with those of past researchers.

6 CONCLUSIONS

An object-oriented evolutionary algorithm has been described. The OO representation of design variables has been shown to exhibit slightly different overall behavior than genetic algorithms using binary strings for design variable representation. Two types of crossover have been discussed and applied within the context of the object-oriented representation. A short empirical study illustrated that object-oriented EA's may require larger populations and higher mutation rates that the corresponding binary GA.

An optimization problem implementing advanced analysis based design assessment was formulated. Discussion of the penalties needed for the inelastic analysis based design process was provided. A two story, three bay steel frame was designed using the object oriented evolutionary algorithm with fully restrained and partially restrained connections. The results provided illustrate that the object oriented EA can achieve designs consistent with those obtained by past researchers.

Acknowledgments

This research was made possible through a grant from the National Science Foundation (CMS 9813216). The authors would also like to acknowledge the fruitful discussions related to genetic algorithms with Mark S. Voss.

References

Bjorhovde, R., Colson, A., Brozzetti, J. (1990). Classification System for Beam-to-Column Connections, *Journal of Structural Engineering*, **116** (11), American Society of Civil Engineers, pp. 3059-3076.

Bridge, R.Q., Clarke, M.J., Osterreider, P., Pi, Y.L., Trahair, N.S. (1998). Design by Advanced Analysis, *Journal of Constructional Steel Research*, **46** (103), Elsevier Science Publishers, Paper 144.

Camp, C., Pezeshk, S., Cao, G. (1998). Optimized Design of Two-Dimensional Structures Using a Genetic Algorithm, *Journal of Structural Engineering*, American Society of Civil Engineers, **119** (5), pp. 551-559.

Foley, C.M. and Vinnakota, S. (1999). Inelastic Behavior of Multistory Partially Restrained Frames – Part I, *Journal of Structural Engineering*, **125** (8), American Society of Civil Engineers, pp. 854-861.

Jenkins, W.M. (1991). Towards Structural Optimization Via the Genetic Algorithm, *Computers & Structures*, **40** (5), Pergamon Press, plc, pp. 1321-1327.

LRFD (1999). *Load and Resistance Factor Design Specification for Structural Steel Buildings*, American Institute of Steel Construction, Chicago, IL, December.

Parmee, I.C. (1995). Diverse Evolutionary Search for Preliminary Whole System Design, In *Developments in Neural Networks and Evolutionary Computing for Civil and Structural Engineering*, Topping, B.H.V. (ed.) CIVIL-COMP PRESS, Edinburgh, UK, pp. 199-204.

Pezeshk, S., Camp, C.V., Chen, D. (2000). Design of Nonlinear Framed Structures Using Genetic Optmization, *Journal of Structural Engineering*, American Society of Civil Engineers, **126** (3), pp. 382-388.

Rivard, H. and Fenves, S.J. (2000). A Representation for Conceptual Design of Buildings, *Journal of Computing in Civil Engineering*, **14** (3), American Society of Civil Engineers, pp. 151-159.

Schinler, D. (2001). *Design of Partially Restrained Steel Frames Using Advanced Analysis and an Object Oriented Evolutionary Algorithm*, MS Thesis, Marquette University, Milwuakee, WI.

SSRC (1988). Technical Memorandum No. 5, *Guide to Stability Criteria for Metal Structures, 4th Edition*, Galambos, T.V. (ed.), John Wiley & Sons, Inc, pp. 732-734.

Voss, M.S. and Foley, C.M. (1999). Evolutionary Algorithm for Structural Optimization, In Banzhaf, W., *et.al.* (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference – Volume 1*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, pp. 678-685.

White, D.W. (1993). Plastic Hinge Methods for Advanced Analysis of Steel Frames, *Journal of Constructional Steel Research*, **24** (2), Elsevier Science Publishers, pp. 121-152.

Wu, A.S., Lindsay, R.K., Riolo, R.L. (1997). Emprical Observations on the Roles of Crossover and Mutation, In Bäck, T. (ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, pp. 362-369.

Xu, L., Sherbourne, A.N., Grierson, D.E. (1995). Optimal Cost Design of Semi-Rigid, Low-Rise Industrial Frames, *Engineering Journal*, **32** (3), American Institute of Steel Construction, pp. 87-97.

Yura, J.A., Galambos, T.V., Ravindra, M.K. (1978). The Bending Resistance of Steel Beams, *Journal of the Structural Division*, **104** (ST9), pp. 1355-1370.

GECCO - 2001

2001 GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE WORKSHOP PROGRAM

San Francisco, California
July 7, 2001