EMPIRICAL ANALYSIS OF MEMETIC ALGORITHMS FOR CONCEPTUAL DESIGN OF STEEL STRUCTURAL SYSTEMS IN TALL BUILDINGS

Rafal Kicinger and Tomasz Arciszewski
Department of Civil, Environmental and Infrastructure Engineering,
George Mason University,
Fairfax, VA, USA
E-mail: rkicinge@gmu.edu

Abstract

This paper discusses the results of extensive design experiments in which memetic algorithms were applied to optimize topologies of steel structural systems in tall buildings. In these experiments, evolutionary algorithms were employed to determine optimal configurations of structural members (topology optimization) while the optimal cross-sections of members (sizing optimization) were found using continuous/discrete optimization algorithm implemented in SODA. The impact of all major evolutionary computation parameters on the performance of memetic algorithms was investigated. Two classes of complex structural design problems were considered: design of a wind bracing system in a tall building and design of the entire steel structural system in a tall building. The total weight of the structural system was assumed as the optimality criterion with respect to which the designs were optimized while satisfying all design requirements specified by appropriate design codes.

In the conducted experiments various key EC parameters and their values were considered having the largest impact on the performance of memetic algorithms. It was discovered that the type of EA, the rate of mutation operator, and the size of parent population were critical for the success of structural optimization processes. Specifically, evolution strategies produced on average significantly better results than genetic algorithms for the design problems considered in the paper. Also, low mutation rates, i.e. 0.025, resulted in best performance of memetic algorithms. Furthermore, small parent population sizes were generally preferred to large populations. For the simpler problem of conceptual design of a wind bracing system, optimal results were produced even when the population with a single member was used. In the case of the second and more complex design problem slightly larger population sizes were required consisting of 5 members.

Results of a large number of design experiments allowed formulating initial recommendations regarding optimal parameter settings for memetic algorithms for structural design applications. The experiments also produced a body of structural design knowledge, both quantitative and qualitative in nature. They identified regions of the design spaces in which high-performance solutions can be found. They also defined the ranges of the total weight of structural systems associated with high-performance solutions for both classes of design problems. Furthermore, significant qualitative differences between high-performance solutions have been identified. The structural shaping patterns exhibited by high-performing designs ranged from crossed macrodiagonal patterns composed of X bracings to irregular patterns consisting of various types of bracings.

Keywords: Memetic algorithms, evolutionary computation, structural design, conceptual design, tall buildings

1. Introduction

Evolutionary computation (EC) is used for solving many complex problems in science and engineering. It allows conducting robust optimization and at the same time has modest requirements on the formulation of the problem to be solved. For example, it does not require continuous variables, differentiable objective functions, etc. Thus, it can be applied to many structural design problems, particularly those with discrete or symbolic variables, or objective functions with nonlinear and stochastic components as is frequently the case in conceptual design.

Parallel search conducted by a population (superset) of solutions is one of the key characteristics of evolutionary algorithms (EAs). It facilitates global exploration of the search space and helps EAs

M. Pandey et al. (eds), Advances in Engineering Structures, Mechanics & Construction, 277–288.
© 2006 *Springer. Printed in the Netherlands.*

escape local optima. In design spaces, these local optima frequently correspond to already known design concepts. Parallel search, however, also has an adverse impact on the ability of EAs to efficiently refine near-optimal solutions. EAs are usually not as good as local search algorithms in converging to the optimal solution once the optimal region of the search space has been found.

Thus, several researchers proposed hybrid algorithms combining excellent global exploration characteristics of EAs and efficient refinement capabilities of local search algorithms (Hoeffler et al., 1973; Moscato, 1989). These hybrid algorithms are called memetic algorithms (MAs) but other names like hybrid EAs, or Lamarckian EAs, have also been used in the literature. Even though several applications of MAs to structural design have been reported, e.g., (Hoeffler et al., 1973; Quagliarella and Vicini, 1998; Sakamoto and Oda, 1993), none of them had such a broad empirical scope as the computational studies described in this paper.

This paper discusses the results of extensive design experiments in which MAs were applied to optimize topologies and cross-sections of steel structural systems in tall buildings. Two classes of complex structural design problems were considered: conceptual and detailed design of a wind bracing system in a tall building and conceptual and detailed design of the entire steel structural system in a tall building. In these experiments, evolutionary algorithms (EAs) were employed to determine optimal configurations of structural members (conceptual design) while the sizing optimization (detailed design) was conducted using continuous/discrete optimization algorithm implemented in SODA (Grierson, 1989). The impact of all major evolutionary computation (EC) parameters on the performance of design processes was investigated. The following parameters were tested: the type of an evolutionary algorithm, parent and offspring population sizes, the type of the generational model, crossover and mutation rates, and the length of design processes (number of fitness evaluations).

The remainder of the paper is organized as follows. First, a brief description of MAs is provided. Next, experimental settings used in the computational experiments are presented. Furthermore, results of extensive computational studies are reported and grouped with respect to experimental parameters being tested. Finally, research conclusions are provided, including recommendations for optimal experimental settings for structural design experiments utilizing MAs as well as directions of future research.

2. Memetic Algorithms

Memetic algorithms are hybrid algorithms in which EAs are integrated with a local search. In these algorithms, EAs are usually used for global exploration of the search space while fine-tuning of solutions is conducted by the local search. MAs have largely emerged in the late 1980s and early 1990s (Moscato, 1989) with the progress in the field of evolutionary computation and improved understanding of the dynamics of evolutionary algorithms. Some earlier studies on hybrid optimization methods utilizing EAs were also reported in the literature, e.g., a hybrid of evolution strategies (Rechenberg, 1973) and linear programming for topology optimization of trusses (Hoeffler et al., 1973).

The overall structure of a canonical MA is presented in Figure 1. It is a slightly modified version of a simple MA presented in (Hart et al., 2005a). Figure 1 shows that the differences between EAs and MAs occur at point 6. In the case of a MA, additional local search is conducted to improve (fine-tune) new individuals generated by an EA in steps 3-5.

This canonical algorithm structure can be instantiated in many different ways. Depending on the type of EA used (e.g., a genetic algorithm (GA) (Holland, 1975), evolution strategies (ES) (Rechenberg, 1973), etc.) and the type of local search (e.g., steepest ascent/descent, greedy search, etc.) different hybrids can be created. In this way, each component of the MA can be adjusted to a specific problem, or a class of problems. Current state-of-the-art review in this field can be found in (Hart et al., 2005b).

MAs have also been applied to solve several structural design problems. Hoeffler et al. (1973) combined linear programming and ES to optimize topologies of truss systems. Linear programming methods were applied to identify initial truss layout and ES were subsequently used to determine optimal positions of joints. This hybrid strategy resulted in reduced total weight of trusses when compared to results obtained using sole linear programming methods. Sakamoto and Oda (1993)

```
1. Initialize the population
2. Evaluate all members of the population
While the termination condition is not satisfied
{
    3. Select individual(s) in the population to be parent(s)
    4. Create new individuals by applying the variation operators to the copies of parent(s)
    5. Evaluate new individuals
    6. Improve individuals using local search
    7. Replace some/all of the individuals in the current population with the new individuals
}
```

Figure 1: Structure of a canonical memetic algorithm

applied a MA composed of a GA and the generalized optimality criterion method to optimize topologies and cross-sections of members in trusses. As before, minimum weight truss systems were sought subject to displacement constraints. At about the same time, Adeli and Cheng (1993) proposed a hybrid algorithm for the optimization of space truss structures. In this case, a GA combined with penalty-function method produced optimal solutions faster than a standard GA.

In the experiments reported in this paper, evolutionary algorithms were combined with a continuous/discrete optimization algorithm implemented in SODA (Grierson, 1989) to optimize topologies and cross-sections of members in steel structural systems of tall buildings. Extensive sensitivity analyses of evolutionary computation parameters were also conducted to determine their optimal settings for structural design experiments.

3. Experimental Design

3.1. Structural Systems in Tall Buildings

Computational experiments with MAs were conducted for two classes of complex structural design problems: design of a wind bracing system (Problem I) and design of an entire steel structural system in a tall building (Problem II). They are illustrated in Figure 2. In Problem I, an optimal configuration of wind bracing elements was sought while keeping the configurations of all other members the same, i.e., all beams, columns, and supports had the same topological configurations in all experiments. On the other hand, an optimal configuration of all structural members of the steel structure in a tall building was sought in Problem II (see Figure 2b). Furthermore, the first problem was further subdivided into the following 3 subproblems:

- Problem Ia - design of a wind bracing system composed of simple X bracings and no bracings (empty cells) only.
- Problem Ib – design of a wind bracing system composed of K bracings and no bracings (empty cells) only.
- Problem Ic – design of a wind bracing system composed of all 7 types of wind bracing elements shown in Figure 3a.

This was motivated by the fact that high-performance solutions for each of these subproblems are not only qualitatively but also quantitatively different (Kicinger, 2004). The impact of these differences on the performance of MAs was investigated. Figure 3b-c represent types of beam and support elements in conducted experiments, respectively. Columns were assumed the same (fixed joints) in all design experiments reported in this paper (for Problems I and II). Hence, there were not evolved by evolutionary algorithms but their cross-sections were optimized by the local search. The optimal cross-sections of structural members were selected from the catalog of standard shapes specified in (American Institute of Steel Construction, 1989).

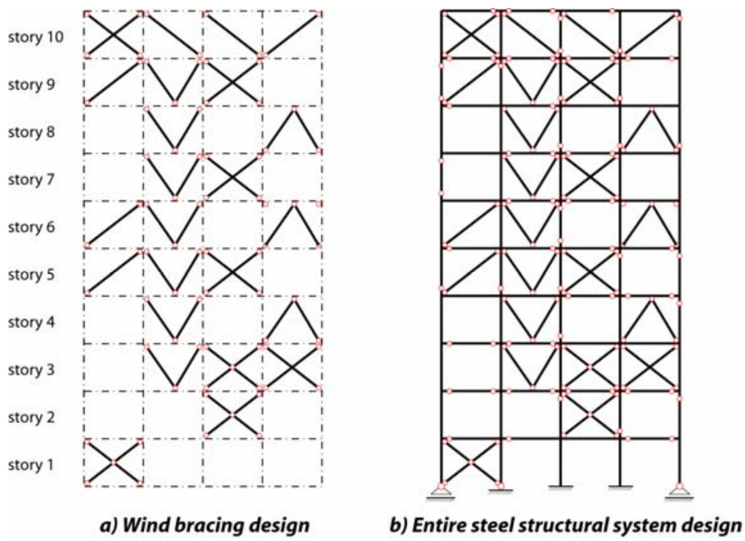

Figure 2: Classes of structural design problems considered in computational experiments

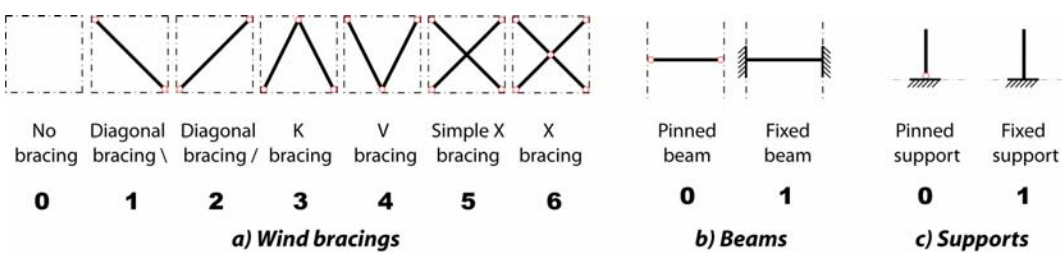

Figure 3: Types of structural members used in the design experiments

The parameters of the design problems and their values are presented in Table 1. It shows that 30-story buildings with 5 bays were considered. The 1^{st} order structural analysis and the local optimization of cross-sections of members were conducted by SODA (Grierson, 1989). SODA is a commercial computer system developed for the analysis, design, and optimization of steel structural systems. Table 2 shows the magnitudes of dead, live, and wind loads assumed in these calculations.

Table 1: Parameters of the design problems

Domain Parameter	Value(s)
Number of stories	30
Number of bays	5
Bay width	20 feet (6.01 m)
Story height	14 feet (4.27 m)
Distance between transverse systems	20 feet (6.01 m)
Structural analysis method	1^{st} order
Beams	pinned, fixed (Problem II)
Columns	fixed
Supports	pinned, fixed (Problem II)
Wind bracings	no, diagonal (/), diagonal (\), K, V, simple X, and X

Table 2: Magnitudes of dead, live, and wind loads

Load Parameter	Value(s)
Dead load magnitude	50 psf (2.39 kN/m2)
Live load magnitude:	
- building	100 psf (4.78 kN/m2)
- roof	30 psf (1.43 kN/m2)
Wind load:	
- Wind speed	100 mph (160.9 km/h)
- Wind importance factor	1.0
- Wind exposure category	C

3.2. Design Representations

In the reported experiments, steel structural systems were represented by linear genomes. For Problems Ia, Ib, and Ic the genomes were homogenous, i.e., they were composed of identical genes encoding wind bracing elements. For Problems Ia and Ib these genes had 2 values (binary representations) and for Problem Ic the genes had 7 values (integer-valued representation). Entire steel structural systems in tall buildings considered in Problem II were represented by nonhomogeneous genomes. They included 3 types of genes:

- encoding wind bracing elements with 7 possible values (see Figure 3a)
- encoding beam elements with binary values (see Figure 3b)
- encoding supports with binary values (see Figure 3c)

The length of the genome depended on the problem considered. For Problems Ia-Ic the genome was 150 genes long whereas for Problem II it consisted of 306 genes.

3.3. Evolutionary Computation Parameters

The major goal of the reported research is determination of optimal experimental settings for MAs applied to complex structural design problems. In order to achieve it, two classes of design problems were selected (described above) and an extensive evolutionary computation parameter search (sensitivity analysis) was conducted. It involved the following parameters and their values: the length of the design process (specified by the number of fitness evaluations), parent and offspring population sizes, the rate of mutation operator, and the rate of crossover operator.

The experiments were divided into two major groups depending on the termination criterion used in individual runs: short-term experiments (up to 1,000 fitness evaluations) and long-term experiments (up to 10,000 fitness evaluations). This distinction is important from the structural design point of view because evaluations of generated designs are usually very expensive (more than 99% of computational time).

Sensitivity analyses were conducted during short-term processes. The optimal combination of parameters' values found in the short-term processes was subsequently used in the long-term experiments. The performance analysis of MAs was conducted for both short- and long-term experiments. It included the following performance criteria:

- performance improvement of the best design at the end of the experiment compared to the best design from an initial population
- performance improvement of the average design at the end of the experiment compared to the average design from an initial population

Both improvements were measured by the reduction of the total weight of a structural system being designed.

An extensive parameter search was conducted during short-term experiments. For all combinations of parent and offspring population sizes shown in Table 3, a search for optimal rates of mutation and crossover was conducted. In each case, 12 combinations of mutation and crossover rates were considered, i.e. (mutation rate 0.025, crossover rate 0), (mutation rate 0.025, crossover rate 0.2), etc. The design processes were repeated 5 times for each combination of parameter values using a different value of a random seed each time.

Two types of memetic algorithms were tested: MA-GA and MA-ES. The former utilizes GAs while the latter uses ES. Both MAs were investigated in order to determine which produces better results when combined with the local search for the two classes of structural design problems. The initial population of parents was generated randomly for each experiment. As discussed above, the fitness of a design was determined by the total weight of the steel structural system calculated using the first-order structural analysis. Whenever an infeasible design was generated, it was assigned the fitness value of 0. In other words, the death penalty method was used to handle infeasible solutions (Coello Coello, 2002).

Table 3: Evolutionary computation parameters and their values

EC Parameter	Value(s)	EC Parameter	Value(s)
Type of MA	MA-ES, MA-GA	*Initialization method*	random
Pop. sizes (parent, offspring)	(1,5), (1,25), (5,25), (5,125) or (50,250) for MA-ES(μ+λ) (5,25), or (50,50) for MA-GA (5,25) for MA-ES(μ,λ)	*Fitness*	Total weight of the structural system (determined by the 1st-order analysis)
Generational model	Overlapping for MA-ES(μ+λ), Nonoverlapping for MA-ES(μ,λ) and MA-GA	*Constraint handling method*	death penalty (infeasible designs assigned 0 fitness)
Selection (parent, survival)	(uniform stoch., truncation) for MA-ES, (fitness prop., uniform stoch.) for MA-GA	*Termination criterion*	1,000 evaluations (short-term), or 10,000 evaluations (long-term)
Mutation rate	0.025, 0.1, 0.3, or 0.5	*Number of runs*	5 (in each experiment)
Uniform crossover rate	0, 0.2, or 0.5		

4. Experimental Results

This section reports the results of design experiments introduced in the previous section. They are grouped with respect to the parameter being investigated.

4.1. Optimal Rates of Mutation and Crossover Operators

Initial experiments focused on finding the optimal rates of mutation and crossover operators understood here as the rates which produced the best progress of MAs. An extensive parameter search was conducted to determine the optimal rates (see Table 3). Obtained results differed for various types of evolutionary algorithms. Typical results for MA-ES are presented in Figure 4 which shows the average best-so-far fitness values and 95% confidence intervals (vertical lines) calculated using Johnson's modified t test (Johnson, 1978) obtained in a series of design experiments with MA-ES(5+25) for Problem Ia. In these experiments, the rate of uniform crossover was equal to 0.2.

A clear pattern can be identified in Figure 4 regarding the impact of the mutation rate on the fitness of produced designs: lower mutation rates produce better fitness (i.e. lower fitness because it is a minimization problem) of designs produced. This pattern was observed in all design experiments involving MA-ES for various parent and offspring population sizes, and crossover rates, as illustrated in Figure 5. It shows average best-so-far fitness values and corresponding confidence intervals obtained at the end of short-term experiments for Problem Ia. Similar patterns were obtained for Problems Ib-c, and Problem II when ES was used.

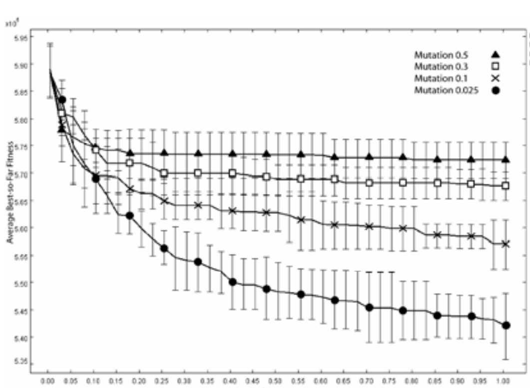

Figure 4: Progress of MA-ES for various mutation rates

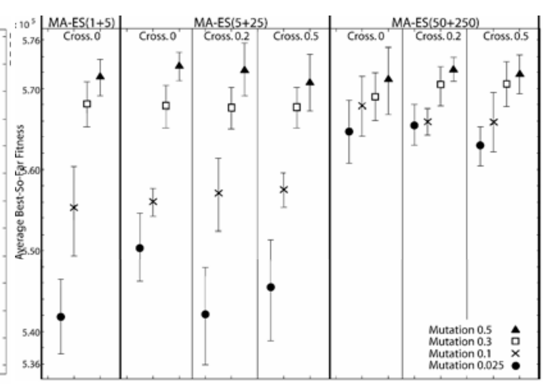

Figure 5: Impact of mutation rates on MA-ES

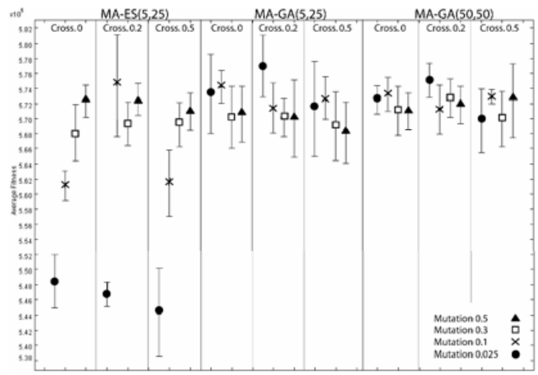

Figure 6: Comparison of the impact of mutation rates on MA-ES and MA-GAs

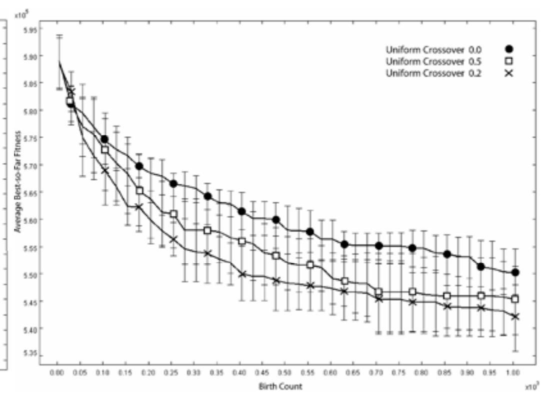

Figure 7: Impact of crossover rates on MA-ES

Different results and patterns were obtained when MA-GAs were employed to optimize topologies of steel structural systems. Figure 6 compares the results produced by two MAs, one employing MA-ES and one utilizing MA-GA. Here, the graphs produced by MA-ES(5,25) (left) are compared to the graphs produced by MA-GA(5,25) (center) and MA-GA(50,50) (right). It is clear that the results produced by MA-GAs are significantly different in terms of patterns produced: higher mutation rates produce better results, particularly when low crossover rates are used. In this case, however, the differences among the results produced by MA-GAs with various rates of mutation are small.

A search for the optimal rate of the crossover operator was conducted by analyzing the results of the design experiments in which various crossover rates were used but the mutation rate was kept the same. Figure 7 presents typical results obtained in the experiments in which MA-ES were used. It shows the average best-so-far fitness values and 95% confidence intervals obtained in the design experiments with MA-ES(5+25) and 3 different rates of crossover, i.e. 0.0, 0.2, and 0.5. The mutation rate was kept the same and equal to 0.025. Figure 7 shows that various crossover rates yielded only small differences in the fitness of produced designs. No clear pattern could be observed, as it was the case with the mutation operator. These observations were further confirmed by the results presented in Figure 8. It shows that there was no trend favoring specific crossover rates. On the contrary, in some cases the best results were achieved with no crossover at all and sometimes the best results were obtained when very high crossover rates are used, i.e. when the rate was equal to 0.5. Figure 8 also shows that even if there were differences among the fitness values obtained with various crossover rates, they were not significant (confidence intervals overlap in all cases). These results were consistent for all design problems reported in the paper. Similarly as in the case of MA-ES, MA-GAs do not exhibit any clear pattern in terms of preferred crossover rates. The graph showing these results was, however, omitted.

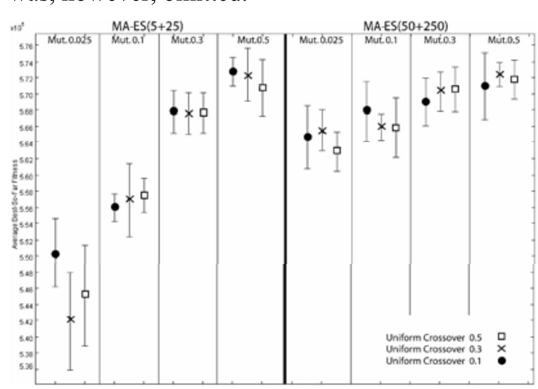

Figure 8: Impact of crossover rates on MA-ES

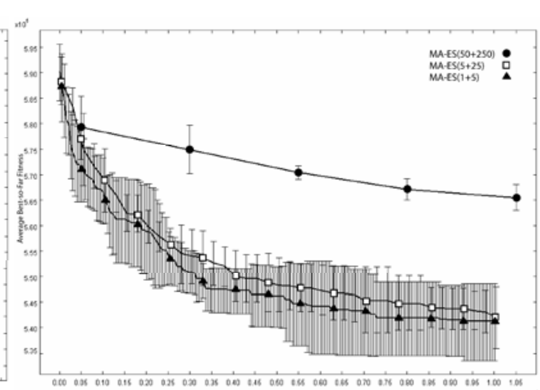

Figure 9: Impact of population sizes on MA-ES

Concluding, MA-ES produce the best results when low rates of mutation operator are used, e.g. 0.025. On the contrary, higher rates of mutation seem to be preferred by MA-GAs but the differences in the obtained results are not as significant as in the case of MA-ES. No such patterns were obtained for crossover rates.

4.2. Optimal Parent and Offspring Population Sizes

The next group of experiments focused on determining the optimal sizes of populations of parents and offspring for MAs. Three different combinations of sizes of parent and offspring populations were considered for MA-ES and two combinations for MA-GAs.

Typical results obtained with MA-ES for Problem I are presented in Figure 9. It shows the results of the design experiments in which three combinations of the parent and offspring population sizes were used, including MA-ES(1+5), MA-ES(5+25), and MA-ES(50+250). Mutation and crossover rates were kept the same in all experiments shown in Figure 9 and equal to 0.025 and 0.2, respectively.

It is clear that MA-ES using large population sizes, i.e. MA-ES(50+250), produced inferior results compared to the other two MA-ES with smaller population sizes. On the other hand, it also produced the smallest variance. The other two MA-ES with smaller population sizes achieved almost the same optimization progress in terms of the average best-so-far fitness of the produced designs. However, MA-ES(1+5), i.e. the 'greedy' MA-ES preserving only the single best individual to the next generation, exhibited much larger variance compared to MA-ES(5+25) which preserves the top 5 individuals to the next generation. Thus, in this case parallel search conducted by MA-ES(5+25) reduces the variance of the obtained results without decreasing the performance of the algorithm. On the other hand, when the size of populations is increased too much, e.g. as in MA-ES(50+250), the reduction of variance comes at a cost of a substantial decrease of the performance of the algorithm.

The outcomes were again different for MA-GAs. In both cases, i.e. for MA-GA(5,25) and MA-GA(50,50), the performance of the algorithm was almost identical. Figure 10 shows typical results of the design experiments involving MA-GA(5,25) and MA-GA(50,50). The specific results presented in this figure were produced by the two algorithms with the same mutation and crossover rates equal to 0.3 and 0.5, respectively.

The two best-so-far curves shown in Figure 10 are almost identical in their nature. The only difference between the two curves is the reduction of variance for the algorithm with larger population sizes, i.e. for MA-GA(50,50). Similar behavior was also observed for MA-ES (see Figure 9). Figure 11 shows that MA-ES produced exactly the same patterns for Problem II as for Problem I. Good optimization progress was obtained for small and medium population sizes, i.e., MA-ES(1+5) and MA-ES(5+25), while the best results in terms of progress and variance were achieved by MA-ES(5+25).

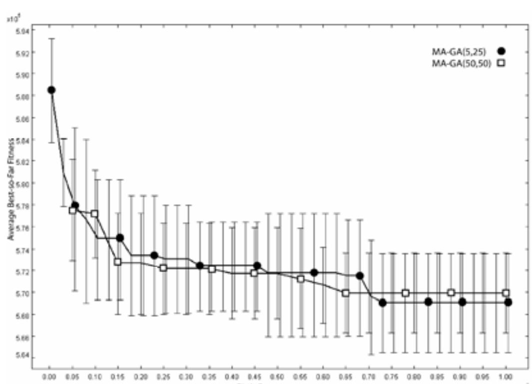

Figure 10: Impact of population sizes on MA-GA

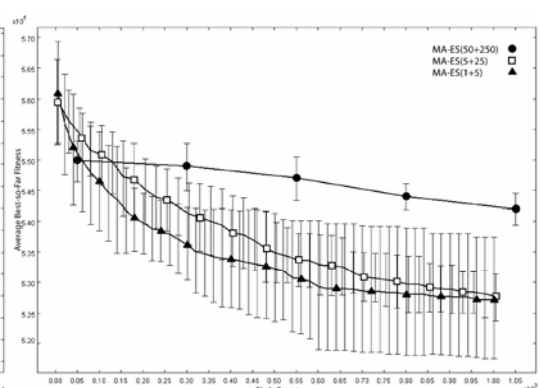

Figure 11: Impact of population sizes on MA-ES for Problem II

Concluding, small population sizes seem to be preferred by MA-ES for these problem domains. However, too small population sizes increase the variance of the obtained results. Good results in terms of both performance and variance were produced when moderate sizes of population sizes were employed, e.g. 5 in the case of the parent population and 25 in the case of the offspring population. The impact of the sizes of parent and offspring populations on the performance of MA-GAs seems to be negligible and related only to the reduction of variance of the obtained results. It didn't influence the actual performance of the algorithms in these problem domains.

4.3. Optimal Generational Model

The type of generational model in EAs determines whether or not best solutions found so far should be kept for the next generation. In the overlapping model, the best solution(s) from the parent population can compete for survival with solutions from the offspring population. On the other hand, in the nonoverlapping model only solutions from the offspring population can compete for survival.

In order to determine the impact of the type of the generational model (overlapping vs. nonoverlapping) on MAs, a series of design experiments involving two kinds of MA-ES was conducted, namely MA-ES(5+25) (overlapping) and MA-ES(5,25) (nonoverlapping). The experiments included a total of 24 design experiments (12 for each algorithm) utilizing all 12 combinations of mutation and crossover rates (see Table 3). Figure 12 shows typical results obtained in these experiments. Here, mutation and uniform crossover rates were equal to 0.025 and 0.2, respectively. Figure 12 shows that there are no significant differences between MA-ES(5,25) and MA-ES(5+25). This type of behavior was observed in all conducted experiments. In several cases MA-ES(5+25) slightly outperformed MA-ES(5,25) but in other cases it produced inferior results. The differences between the two generational models were, however, small both in terms of variance and fitness of the generated designs. Generally, it can be concluded that MA-ES with the overlapping and nonoverlapping generational model produce comparable results in these problem domains.

4.4. Optimal Type of Evolutionary Algorithm

The choice of the type of EA determines the properties of global exploration in MAs. In order to determine which EA produces better results in the investigated problem domains, a series of design experiments was performed. Figure 13 shows a comparison of the behavior of two algorithms, i.e., MA-ES and MA-GA, for Problem Ia. Two average best-so-far curves in the upper part of Figure 13 correspond to the best results obtained with MA-GAs with two combinations of parents and offspring population sizes, i.e. MA-GA(5,25) and MA-GA(50,50). In both cases the mutation rate was equal to 0.3 and crossover rate was equal to 0.5. The results produced by MA-GAs are compared to the average best-so-far performance produced by MA-ES with the overlapping (MA-ES(5+25)) and nonoverlapping (MA-ES(5,25)) generational models. In this case, the rates of mutation and crossover were equal to 0.025 and 0.2, respectively

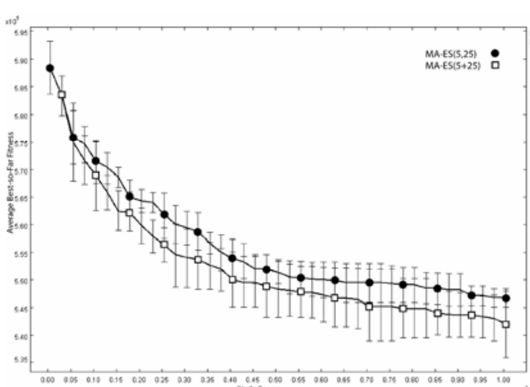

Figure 12: Impact of the generational model on MA

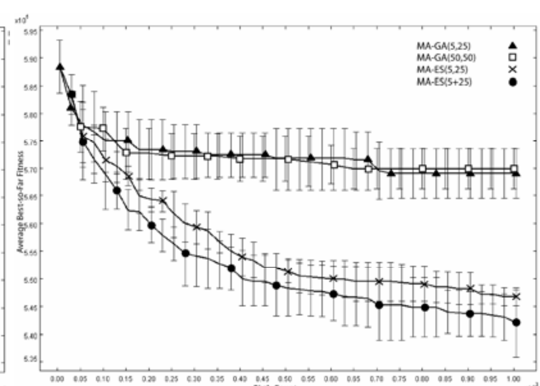

Figure 13: Impact of the type of EA on MA

Figure 13 clearly shows that MA-ES outperformed MA-GA in this problem domain. The average fitness value produced by MA-GA(5,25) after 1,000 evaluations was equal to 569,056 lbs. compared to 542,029 lbs. achieved by MA-ES(5+25). This corresponds to almost 5% better results, on average, produced by MA-ES. The performance improvement between an average design produced after 1,000 fitness evaluations and an average design in the initial population was equal to 19,434 lbs., or 3.3%, for MA-GA(5,25). On the other hand, for MA-ES(5+25) these values were equal to 46,461 lbs. and 7.9%, respectively.

Concluding, the results of the design experiments revealed that MA-ES performed better than MA-GAs in these problem domains. Hence, they were employed in the long-term design experiments reported in the next section.

4.5. Length of Evolutionary Processes

The length of an evolutionary process influences the computational effort required to run a design experiment. This issue is particularly relevant to structural engineering due to the fact that evaluations of structural designs are usually computationally expensive. In yet another series of design experiments with MAs, the impact of the length of design processes (measured by the total number of fitness evaluations) on the quality of produced designs was investigated.

Figure 14 shows the progress of the long-term experiments (10,000 fitness evaluations) with MA-ES for Problem Ic and compares it to the average fitness obtained after 1,000 evaluations (short-term experiment). The average performance improvement between the long-term processes and short-term processes was equal to about 21,900 lbs., or 4.3 percent. The difference between the average fitness after 10,000 fitness evaluations and the average fitness of the initial parents was equal to more than 55,500 lbs., or 10.2 percent. Similar results were produced by MA-ES for Problem II. Figure 15 shows a similar comparison of the long- and short-term experiments. In this case, the average fitness of the designs produced in the long-term experiments equaled 502,879 lbs. and was more than 21,500 lbs., or 4.1 percent, better than the average fitness obtained in the short-term experiments. The overall performance improvement in the long-term experiments was, on average, equal to more than 58,000 lbs., or 10.3 percent, compared to about 36,000 lbs., or 6.4 percent, achieved in the short-term optimization experiments.

The long-term experiments took ten times more computational time than the short-term experiments. Considering the fact that each evaluation took about 1 minute on a Pentium IV processor, the total time required for a single run of a long-term experiments was equal to almost 7 days compared to about 16 hours necessary for a short-term experiment. Thus, there is always a strong trade-off between the length of a design process and available computational resources. In this case, we can utilize the fact that after a certain number of fitness evaluations the best-so-far fitness curves tend to level-off (see Figure 15) and significant performance improvements are no longer obtained. This threshold, however, needs to be defined empirically for individual problems.

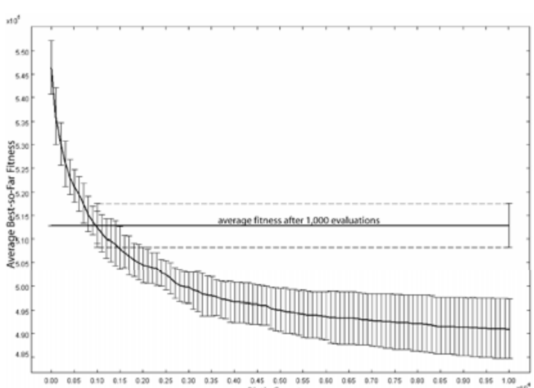

Figure 14: Impact of the length of design process on performance of MA-ES for Problem Ic

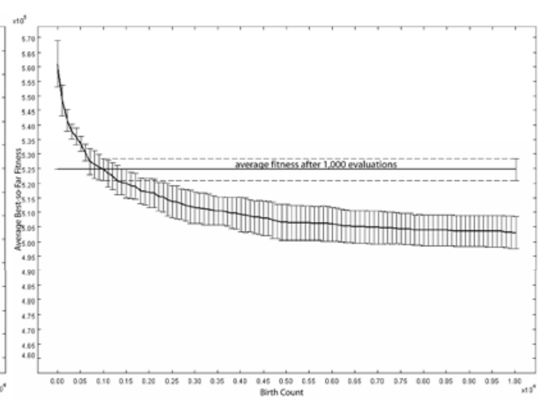

Figure 15: Impact of the length of design process on performance of MA-ES for Problem II

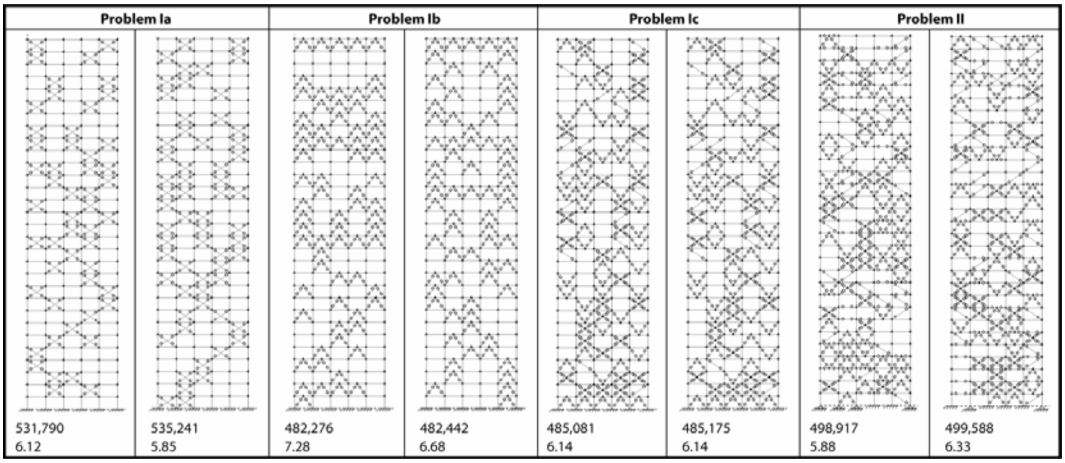

Figure 16:Best designs produced in design experiments with MAs for each problem

4.6. Optimal Designs

In extensive design experiments, (sub)optimal designs were produced for each class of structural problems and subproblems investigated in the paper. The final designs generated by MAs for each problem differed not only in the total weight but also in structural shaping patterns produced by configurations of wind bracings. The best designs produced in these experiments are shown in Figure 16. The values located below designs indicate their total weight in lbs. (top) and their maximum horizontal displacement (bottom) in inches.

Figure 16 shows significant differences in fitness (total weight) corresponding to best solutions found for each problem. For example, steel structural systems with the wind bracing system composed solely of X bracing elements (Problem Ia) have generally higher total weight by about 40,000-50,000 lbs. than best designs for other problems. On the other hand, they typically exhibit smaller horizontal displacements (better stiffness). They also show emerging structural shaping pattern of crossed macrodiagonal bracings in the lower and middle part of the structural system (see Figure 16). On the contrary, best solutions for Problems Ib-c and II exhibit random-looking configurations of wind bracing elements.

5. Conclusions

This paper reported results of a large number of design experiments with memetic algorithms. It also formulated initial recommendations regarding optimal parameter settings for these algorithms when applied to complex structural design problems. These heuristics can be used by researchers and practitioners to quickly set up their design experiments without conducting expensive parameter optimization.

In the reported studies, key evolutionary computation parameters and their values were identified. It was discovered that the type of evolutionary algorithm, the rate of mutation operator, and the size of the parent population were critical for the success of structural design processes. Specifically, evolution strategies produced on average significantly better results than genetic algorithms for the design problems considered in the paper. Also, low mutation rates, i.e. 0.025, resulted in better performance of memetic algorithms. Furthermore, small parent population sizes were generally preferred to large populations.

The experiments also produced significant body of structural design knowledge, both quantitative and qualitative in nature. Specifically, they identified regions of the design spaces in which high-performance solutions could be found. They also determined the ranges of the total weight of structural systems associated with high-performance solutions for both classes of design problems and showed that these ranges are significantly different. Finally, qualitative differences between high-performance solutions have been identified in terms of structural shaping patterns exhibited by configurations of wind bracing elements.

The research presented in this paper will be continued, including the extension of the scope of the empirical studies to other structural design problems. Also, other local search algorithms will be combined with evolutionary algorithms and applied to several structural engineering problems. Another promising direction of future research includes more advanced representations of structural systems and their impact on the performance of memetic algorithms.

References

Adeli, H., and Cheng, N. T. (1993). "Integrated genetic algorithm for optimization of space structures." *Journal of Aerospace Engineering*, 6(4), 315-328.

American Institute of Steel Construction. (1989). *AISC Manual of Steel Construction*, Ninth edition, AISC, Chicago, IL.

Coello Coello, C. A. (2002). "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art." *Computer Methods in Applied Mechanics and Engineering*, 191, 1245-1287.

Grierson, D. E. (1989). "Computer-automated optimal design for structural steel frameworks." *Proceedings of the NATO ASI Conference on Optimization and Decision Support Systems in Civil Engineering*, B. H. V. Topping, ed., Edinburgh, UK, 327-354.

Hart, W. E., Krasnogor, N., and Smith, J. E. (2005a). "Memetic evolutionary algorithms." Recent advances in memetic algorithms, W. E. Hart, N. Krasnogor, and J. E. Smith, eds., Springer-Verlag, Berlin Heidelberg, 3-27.

Hart, W. E., Krasnogor, N., and Smith, J. E. (2005b). *Recent advances in memetic algorithms*, Springer-Verlag, Berlin Heidelberg.

Hoeffler, A., Leysner, U., and Weidermann, J. (1973). "Optimization of the layout of trusses combining strategies based on Mitchel's theorem and on biological principles of evolution." *Proceedings of the 2nd Symposium on Structural Optimization*, Milan, Italy.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, Michigan.

Johnson, N. J. (1978). "Modified t tests and confidence intervals for asymmetrical populations." *Journal of the American Statistical Association*, 73(363), 536-544.

Kicinger, R. (2004). "Emergent Engineering Design: Design creativity and optimality inspired by nature," Ph.D. Dissertation, School of Information Technology and Engineering, George Mason University, Fairfax, VA, USA.

Moscato, P. (1989). "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms." *Caltech Concurrent Computation Program Report 826*, California Institute of Technology, Pasadena, CA, USA.

Quagliarella, D., and Vicini, A. (1998). "Coupling genetic algorithms and gradient based optimization techniques." Genetic algorithms and evolution strategies in engineering and computer science: recent advances and industrial applications, D. Quagliarella, J. Periaux, C. Poloni, and G. Winter, eds., John Wiley & Sons, Chichester, England.

Rechenberg, I. (1973). *Evolutionsstrategie; Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, Stuttgart-Bad Cannstatt.

Sakamoto, J., and Oda, J. (1993). "Technique for optimal layout design for truss structures using genetic algorithms." *Proceedings of the 34th AIAA/ASCE/ASME/AHS Structural Dynamics and Material Conference AIAA/ASME Adaptive Structures Forum*, New York, NY, 2402-2408.