Parameterized versus Generative Representations in Structural Design: An Empirical Comparison

Rafal Kicinger
George Mason University
4400 University Drive MS 4A6
Fairfax, VA 22030
+1 703-993-1658
rkicinge@gmu.edu

Tomasz Arciszewski
George Mason University
4400 University Drive MS 4A6
Fairfax, VA 22030
+1 703-993-1513
tarcisze@gmu.edu

Kenneth De Jong
George Mason University
4400 University Drive MS 4A5
Fairfax, VA 22030
+1 703-993-1553
kdejong@gmu.edu

ABSTRACT
Any computational approach to design, including the use of evolutionary algorithms, requires the transformation of the domain-specific knowledge into a formal design representation. This is a difficult and still not completely understood process. Its critical part is the choice of a type of design representation. The paper addresses this important issue by presenting and discussing results of a large number of design experiments in which parameterized and generative representations were used. Particularly, their computational and design related advantages and disadvantages were investigated and compared.

Evolutionary design experiments reported in this paper considered two classes of structural design problems, including the design of a wind bracing system and the design of an entire structural system in a tall building. Parameterized and generative representations of the structural systems were introduced and their basic features discussed. The generative representations investigated in the paper were inspired by the processes of morphogenesis occurring in nature. Specifically, one-dimensional cellular automata were used to develop, or 'grow,' structural designs from the corresponding 'design embryos.'

The conducted research led to three major conclusions. First, generative representations based on cellular automata proved to scale well with the size of the considered design problems. Second, generative representations outperformed parameterized representations in minimizing weight of the structural systems in our problem domain by generating better designs and finding them faster. Finally, extensive experimental studies showed significant differences in optimal settings for evolutionary design experiments for the two representation types. The rate of mutation operator, the size of the parent population, and the type of the evolutionary algorithm were identified as the evolutionary parameters having the largest impact on the performance of evolutionary design processes in our problem domain.

Categories and Subject Descriptors
J.2 [**Computer Applications**]: Physical Sciences and Engineering – *Engineering*

General Terms
Algorithms, Performance, Design.

Keywords
Design Representations, Evolutionary Computation, Generative Representations, Structural Design, Cellular Automata.

1. INTRODUCTION
Appropriate choice of a design representation is one of the most important and difficult aspects of any evolutionary design (ED) application. Although traditionally employed parameterized representations have proved to perform well for many structural optimization problems, they are considered inadequate when novel designs are sought. They have also experienced some scaling-up issues as the design application problems increased in size and complexity.

For this reason, alternative ways of representing engineering systems have been proposed [3, 4, 9]. These representations, contrary to the parameterized representations, do not encode complete design concepts but rather rules on how to develop, or 'grow,' these designs. It has been shown that these generative representations improve the scalability of ED systems [5] and produce novel designs exhibiting interesting and qualitatively different patterns from known designs [6].

There are still, however, many unanswered questions regarding comparative advantages of both approaches. It is not clear for what types of problems generative representations should, or should not, be used. Similarly, we still don't know much about 'optimal' evolutionary computation (EC) settings for generative systems and their relationship to the best EC parameter values used in traditional ED applications. This paper presents an empirical comparison of these two types of representations in the context of two complex structural design problems. It also provides initial recommendations on how to choose the appropriate type of encoding and its preferred values of EC parameters.

2. BACKGROUND
2.1 Evolutionary Design Representations
Selection of the best type of design knowledge representation and the development of a problem specific encoding are one of the key elements of any computational design activity. A choice of a particular representation of an engineering system is highly influenced by the designer's goal, i.e., whether the emphasis is on optimality in terms of numerical values in the context of a specific design concept, or on the generation of novel design concepts.

Design representations in conceptual design are usually expressed in terms of symbolic attributes (they take values from an unordered or partially ordered set). On the other hand, numerical attributes are used mostly in the detailed design stage [2]. Traditionally, structural design applications were focused strictly on design optimization issues. Hence, the vast majority of applications of evolutionary methods used simple parameterizations of engineering systems encoded in terms of binary, real, or integer-valued attributes [7].

Recently, there has been a growing interest in applying evolutionary methods to conceptual design problems in which one of the important objectives is the development of *novel* designs of structural systems. This has been coupled with significant research efforts within the EC community focused on studying alternative ways of representing designs. Several researchers investigated generative representations which do not encode complete design concepts but rather rules which determine how to construct these designs [3, 5].

As a result of this research, several novel ways of encoding structural designs have been proposed, including Voronoi-based representations and IFS representations based on fractal theory [4], tree-based representations [11], and cellular automata [9]. The state-of-the-art of evolutionary computation and design representations in structural engineering can be found in [7].

2.2 Structural Systems in Tall Buildings

Design of steel skeleton structures in tall buildings is considered as one of the most complex structural engineering problems. These types of structures are usually designed as a system of vertical members called *columns*, horizontal members called *beams*, and various diagonal members called *wind bracings*, since they are added to brace columns and beams in order to increase the flexural rigidity of the entire system.

Skeleton structures are designed to provide a structural support for tall buildings. They have to satisfy numerous requirements regarding the building's stability, transfer of loads (gravity and wind), deformations, etc. For this reason, the design of structural systems in tall buildings requires the analysis of their behavior under various combinations of loading and the determination of optimal configurations (topologies) of structural members, i.e., configurations of beams, columns, wind bracings, and supports.

The two design problems considered in this paper represent two classes of conceptual design problems characterized by distinct levels of their structural complexity. First, a relatively simpler problem of designing a wind bracing system in a tall building is investigated. In this problem, an optimal topology of only wind bracing members is sought assuming the same *configurations* of beams, columns, and supports throughout an entire design process. In this case, however, cross-sections of *all* members (including beams, columns, and wind bracings) are optimized during the detailed design stage (sizing optimization). Second, a more complex problem of designing entire steel structural systems in tall buildings is investigated in which optimal configurations of all structural members are sought, followed, as before, by the sizing optimization.

3. DESIGN REPRESENTATIONS

In this section, we introduce two types of representations of steel structural systems in tall buildings, i.e., a parameterized representation with integer-valued attributes and a generative representation based on one-dimensional cellular automata.

3.1 Parameterized Representations

In parameterizations of engineering systems, the attributes describing an engineering system are directly encoded as genes and their alleles are evolved using evolutionary algorithms (EAs). This is illustrated in Figure 1, which shows a parameterized representation of a steel structural system (see Figure 1b). In this case, symbolic attributes (representing types of structural members) are encoded directly in a parameterized representation.

The 16-story building with 5 bays shown in Figure 1 is represented by a genome consisting of 166 genes. 80 genes encode attributes representing wind bracing elements (see Figure 2), 80 genes encode beams (see Figure 3), and 6 genes represent supports (see Figure 3). Similarly as in [10], column members of a structural system are not evolved but assumed the same (continuous columns) during the entire evolutionary processes.

In this paper, fixed-length genotypes are used to represent wind bracing systems and entire steel structural systems in tall buildings. The length of a genotype used in a given situation, however, depends on the design problem being studied (a wind bracing system or the entire steel structural system) and on the number of stories and bays in the structural system being considered. For example, the 16-story building with 5 bays (see Figure 1) is represented by a genome of length of 166, while a 30-story building with the same number of bays requires 306 genes. Once the design problem and topological properties of a tall building are determined, then the length of the genotype is completely defined and does not change during the ED process.

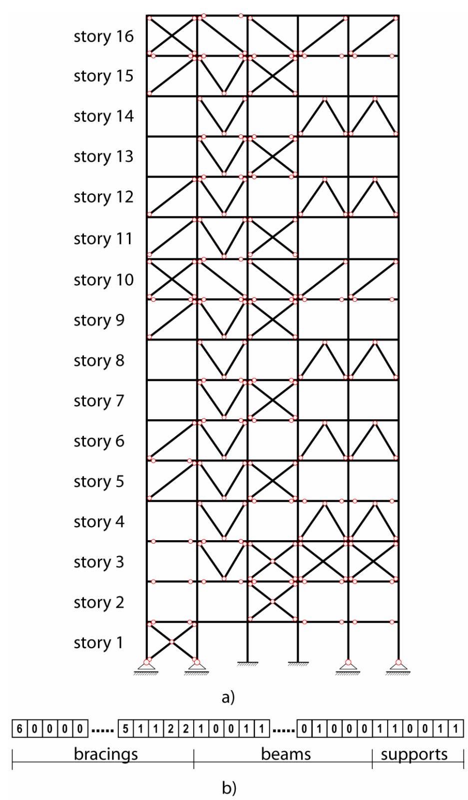

Figure 1. Parameterized representation of a structural system in a tall building

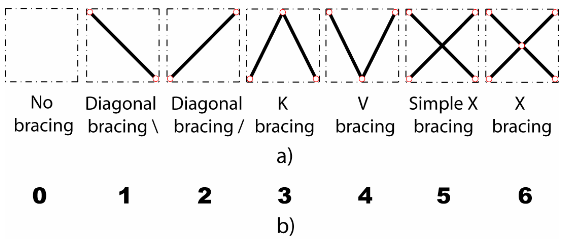

Figure 2. Phenotypic and genotypic representation of wind bracing elements

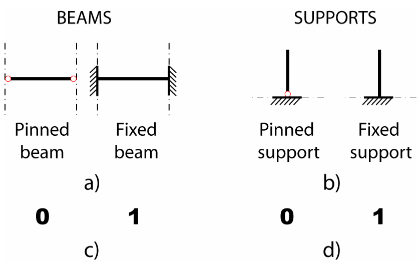

Figure 3. Phenotypic and genotypic representation of beams and supports

3.2 Generative Representations

The generative representations of steel structural systems in tall buildings investigated in this paper have been initially proposed in [8] and [9]. The inspiration for these design representations comes from the processes of *morphogenesis* occurring in nature. Nature manipulates *rules* for growing complex organisms, called 'genetic plans,' rather than the complex organisms themselves. The organisms are then built from the plans via a developmental process (morphogenesis).

The generative representations of steel structural systems investigated in this paper are based on similar principles. Each subsystem of a steel structure, i.e., a subsystem of wind bracings, and a subsystem of beams, is developed from its initial 'design embryo.' Thus, the complete configuration of all members of a given subsystem is not directly encoded in the representation (as in the parameterized representation) but rather it is gradually built by applying a 'design rule' to its corresponding design embryo (a developmental process).

Figure 4a provides a schematic overview of the structure of a genome. It consists of 5 distinct parts. The first two parts encode the design embryo and the design rule which generate the subsystem of wind bracings. The following two parts encode the design embryo and the design rule, which build the subsystem of beams. Finally, the last part of the genome encodes the configuration of supports. In this case no design rule is necessary because the configuration of supports is one-dimensional and it is fully determined by its design embryo.

Figure 4b illustrates the process of development of a steel structural system in a tall building from its generative representation. Specifically, the configurations of a wind bracing subsystem and a beam subsystem are generated from their corresponding design embryos. The design embryos specify the configurations of first stories of the wind bracing subsystem and the beam subsystem, respectively. The configurations of subsequent stories are determined by the iteration of the corresponding design rules which are based on 1D cellular automata (CAs) [12].

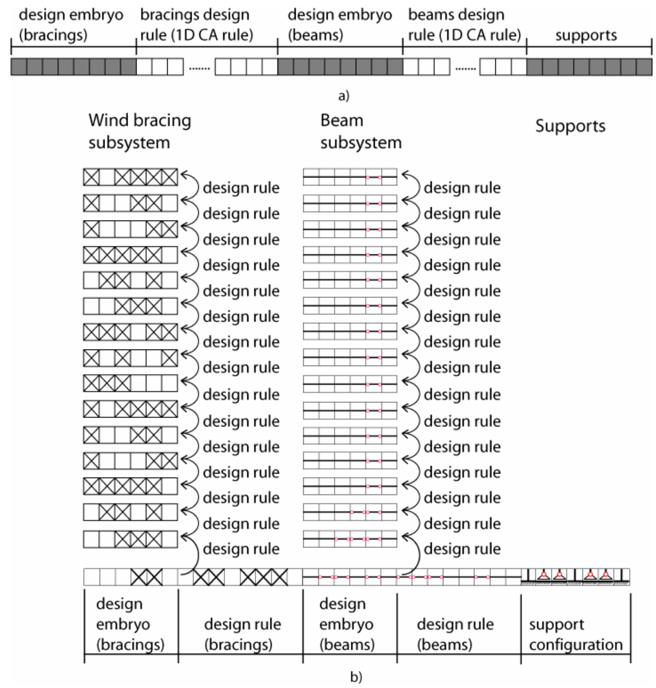

Figure 4. Generative representation of a structural system in a tall building based on multiple 1D CAs

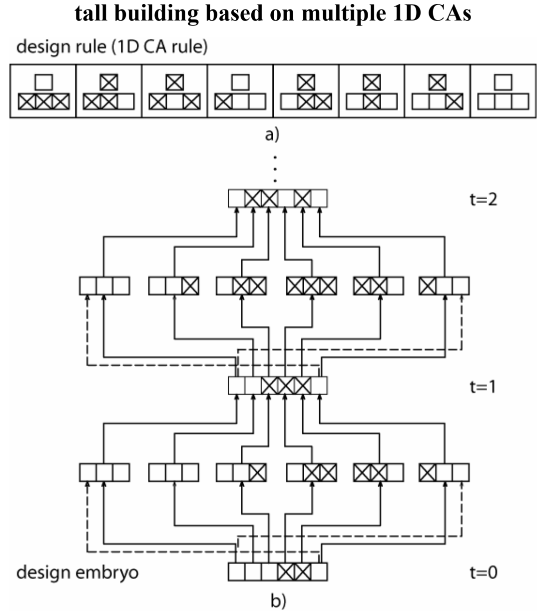

Figure 5. Graphical representation of a design rule based on 1D CA rule and several steps of a developmental process

A graphical illustration of a design rule based on 1D CA rule and initial steps of the developmental process during which a design of a wind bracing subsystem is built are presented in Figure 5. It shows the structure of the design rule encoded in the second part of genome (encoding of the design rule in Figure 4b which develops the bracing subsystem). The bottom row of cells shows all possible combinations of the local neighborhoods of size 3 with 2 types of wind bracing elements (empty cell representing no bracing and X bracing). The top row of cells in Figure 5a presents the outcome values of the 1D CA rule, i.e., the values attained by the central cell in a given local neighborhood at the next time step (here at the next story in a tall building). If we assume a fixed ordering of the local neighborhoods then we can represent any design rule based on 1D CA rule by its outcome values. Hence, the design rule for the wind bracing subsystem (the second part of genome in Figure 4b), can be fully represented by 8 outcome values shown in the top row in Figure 5a.

Several initial steps of the developmental process during which a design of a wind bracing subsystem is gradually built are presented in Figure 5b. Bottom part of Figure 5b (t=0) shows the configuration of the design embryo (first part of the genome shown in Figure 4b). The process of developing the configuration of the next story of the wind bracing subsystem (t=1) from this design embryo involves several operations. First, a set of local neighborhoods of size 3 is constructed by taking each cell from the design embryo together with its left and right neighbors and placing them respectively in the middle, left, and right of the lattice defining each local neighborhood (see the set of 6 local neighborhoods of size 3 placed above the design embryo in Figure 5b). In this instance, so-called periodic boundary conditions are used, meaning that the rightmost cell in the design embryo becomes the left neighbor of the leftmost cell in the design embryo, and vice versa (denoted by dashed lines in Figure 5b). Second, the local neighborhoods created that way are compared to the local neighborhoods which define the conditions of the design rule (see the bottom row in Figure 5a). When the two match, the value shown in the top row in Figure 5a defines the new value of the central cell in the configuration of the next story of the wind bracing subsystem. This process is repeated for each local neighborhood and the obtained values are placed in appropriate positions of the configuration of the next story (t=1). In this way, the second story is fully defined and at the same one iteration (step) of a design rule is completed. This process is repeated as many times as to define configurations of all stories of the wind bracing subsystem. A more detailed description of this type of representations can be found in [6].

The 16-story building with 6 bays shown in Figure 4 is represented by a genome consisting of 35 genes. In this case, 2 types of wind bracing elements, 2 types of beams, 2 types of supports, and design rules based on elementary CAs (i.e., with binary cell values and the local neighborhood of size 3) are considered. 14 genes encode the design embryo and the design rule generating the wind bracing subsystem, 14 genes encode the design embryo and the design rule generating the beam subsystem, and 7 genes encode the design embryo representing supports. When the number of possible types of wind bracing elements grows to 7 (as in Figure 2), then the length of the genome increases to 370. There is a way to significantly reduce the length of the genome by using design rules based on *totalistic* CAs [12]. In this case, the length of the genome decreases to 42.

4. DESIGN EXPERIMENTS

We chose two complex conceptual design problems in structural engineering for our initial empirical comparison of design representations: design of a wind bracing system (Problem I) and design of an entire steel structural system in a tall building (Problem II). 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 2.

This was motivated by the fact that (sub)optimal solutions for each of these subproblems exhibit dramatically different structural shaping patterns, i.e., the best designs are not only quantitatively and but also qualitatively different [6].

Table 1. Parameters of the design problems and their values

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)
Structural analysis method	1st order
Beams	pinned, fixed (Problem II)
Column	fixed
Supports	pinned, fixed (Problem II)
Wind bracings	no, diagonal (/), diagonal (\), K, V, simple X, and X

The parameters of the design problems and their values are presented in Table 1. It shows that a wind bracing system and an entire steel structural system of a 30-story building with 5 bays were the subject of design. For problems Ia, Ib, and Ic the *topology* (the configuration of structural members) of only a wind bracing system was evolved. On the contrary, for problem II the topologies of subsystems of wind bracings and beams as well as the configuration of supports were optimized. The topology of a column subsystem was not evolved in any case and all column members in steel structures were assumed the same during the entire design process, similarly as it was done in [10].

When the topology/configuration of the wind bracing system (problems Ia-Ic), or the entire steel structural system (problem II), were defined, the *sizing* optimization algorithm implemented in SODA was used to determine the optimal cross-sections of all structural members in the entire steel structural system. SODA is a commercial computer system developed for the analysis, design, and optimization of steel structural systems. In our case, it was used as a behavior simulator which evaluates generated designs.

In the reported research, the sizing optimization was conducted for all structural members, including wind bracings, beams, and columns. The optimal cross-sections of structural members were selected from the catalog of standard shapes specified in [1]. In the 1st-order structural analysis conducted by SODA, standard types of loads, load magnitudes, and types of load combinations were considered as specified by the relevant design codes.

4.1 Representational Characteristics

Table 2 shows the values of the representational parameters used in the design experiments. In the reported research two types of design representations were investigated: parameterized and generative. Moreover, two types of design rules were employed in the case of generative representations: based on standard CAs and totalistic CAs. The radius of the local neighborhood was either equal to 1 or 2 and periodic boundary conditions were used in all design experiments with generative representations. The lengths of the genomes for all types of design representations and all design problems considered in this paper are presented in Table 2.

Table 2. Representation parameters and their values

Representation Parameter	Value(s)
Parameterized:	
- number of gene values	2, or 7
- length	150(Ia, Ib, Ic), or 306(II)
Generative:	
- type of CAs	standard, or totalistic
- radius of the local neighborhood	1, or 2
- CA boundaries	periodic
- number of cell values	2, or 7
- length:	
standard CAs, radius 1	13(Ia, Ib), 348(Ic), or 367(II)
standard CAs, radius 2	37(Ia, Ib), 16812(Ic), or 16855(II)
totalistic CAs, radius 1	9(Ia, Ib), 24(Ic), or 39(II)
totalistic CAs, radius 2	11(Ia, Ib), 36(Ic), or 53(II)

4.2 EA Characteristics

The details of the evolutionary computation parameters are presented in Table 3. It shows that the design experiments were divided into two major groups depending on the termination criteria used in individual evolutionary runs: short-term experiments (1,000 fitness evaluations) and long-term experiments (10,000 fitness evaluations). This distinction is important from the structural design point of view because evaluations of generated design concepts are usually very expensive (more than 99% of computational time).

Extensive sensitivity analysis was conducted during the short-term experiments. They involved the following evolutionary computation parameters: mutation rates, crossover rates, sizes of parent and offspring populations, the type of the generational model, and the type of an evolutionary algorithm. Optimal settings for these parameters were sought and, once found, later utilized in the long-term experiments. The performance analysis of ED processes was conducted for both the short-term and the long-term experiments.

The results produced by the parameterized and generative representations were subsequently compared using two performance criteria:

- Average performance improvement – fitness improvement of an *average* design at the end of an experiment compared to an average design from an initial population
- Best performance improvement – fitness improvement of the *best* design at the end of an experiment compared to the best design from an initial population

5. EXPERIMENTAL RESULTS

We used the experimental settings described above to run a large number of design experiments. Their results, grouped with respect to the parameter being investigated, are described below.

5.1 Mutation and Crossover Rates

Initial experiments focused on finding the optimal rates of mutation and crossover operators understood here as the rates which produced the best progress of ED processes. An extensive parameter search was conducted to determine the optimal rates. It involved 12 combinations of mutation and crossover rates (see Table 3).

Table 3. EC parameters and their values

EC Parameter	Value(s)
EA	ES, GA
Pop. sizes (parent, offspring)	(1,5), (1,25), (5,25), (5,125) or (50,250) for ES (5,25), or (50,50) for GA
Generational model	Overlapping for ES(μ+λ), Nonoverlapping for ES(μ,λ) and GA
Selection (parent, survival)	(uniform stoch., truncation) for ES, (fitness prop., uniform stoch.) for GA
Mutation rate	0.025, 0.1, 0.3, or 0.5
Crossover rate	0, 0.2, or 0.5
Fitness	Total weight of the structural system (determined by the 1st-order analysis)
Initial. method	random
Constraint handling method	death penalty (infeasible designs assigned 0 fitness)
Termination criterion	1000 evaluations (short-term), or 10,000 evaluations (long-term)
Number of runs	5 (in each experiment)

The obtained results differed for the two types of design representations considered. Figure 6 shows typical results obtained in experiments with parameterized representations. Specifically, it presents average best-so-far fitness values and 95% confidence intervals (vertical lines) calculated using Johnson's modified t test and obtained in a series of design experiments (problem Ia) with ES(5+25). In these experiments, the rate of uniform crossover was equal to 0.2.

In Figure 6 a clear pattern can be identified regarding the impact of the mutation rate on the fitness of produced design concepts: lower mutation rates produce better fitness (i.e., lower because it is a minimization problem) of design concepts. This pattern was observed in all design experiments involving ES with various parent and offspring population sizes, and crossover rates, as it is illustrated graphically in Figure 7.

Figure 7 clearly shows that the best performance of ES(μ+λ) in the short-term ED processes with parameterized representations was obtained for the lowest mutation rate equal to 0.025. The same pattern was observed for all design problems considered in this paper, i.e., Ia-Ic, and II.

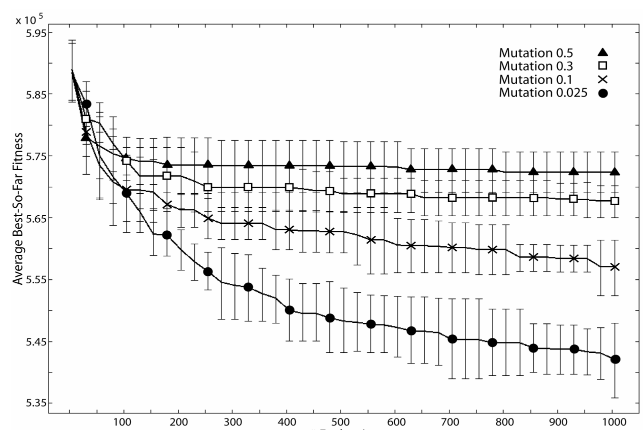

Figure 6. Impact of the mutation rate on the progress of evolutionary processes with parameterized representations

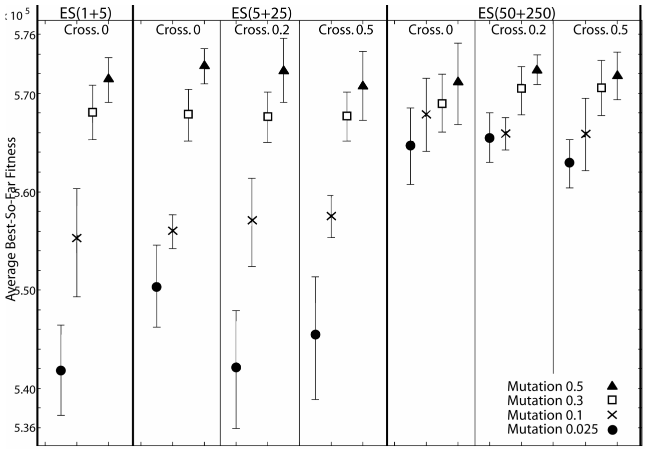

Figure 7. Average fitness obtained after 1,000 evaluations in design experiments with parameterized representations

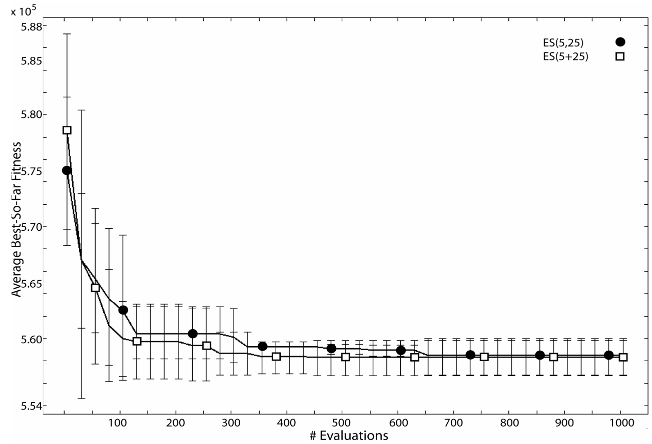

Figure 9. Average fitness obtained after 1,000 evaluations in design experiments with generative representations

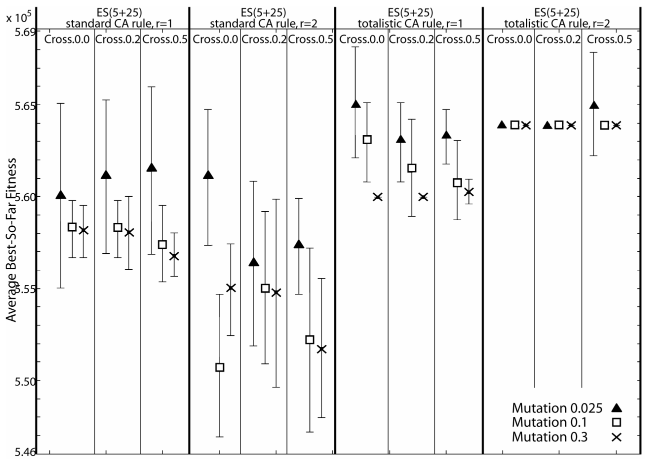

Figure 8. Average fitness obtained after 1,000 evaluations in design experiments with generative representations

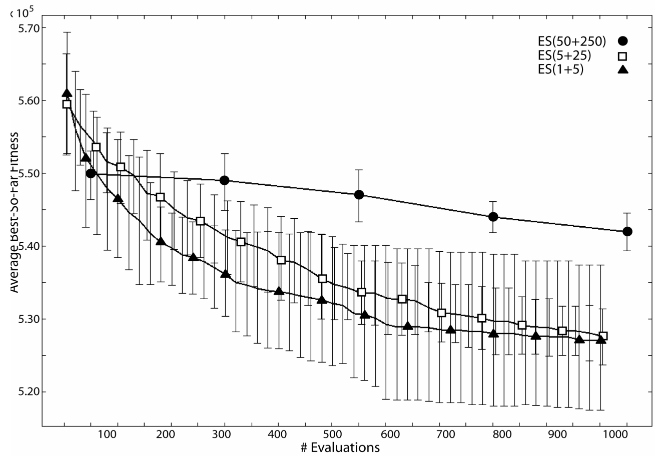

Figure 10. Impact of the population sizes on the progress of evolutionary processes with parameterized representations

The design experiments with generative representations showed, however, a dramatically different pattern. Here, higher mutation rates generally produced better results than lower mutation rates as it is illustrated in Figure 8. It specifically shows average fitness values obtained after 1,000 evaluations for the problem Ia and produced in the ED experiments with ES(5+25) and generative representations based on standard and totalistic CAs with the radius *r* of the local neighborhood equal to 1 and 2.

No such patterns were observed for crossover rates. In some cases the best results were obtained for high rates of crossover operator and sometimes when the crossover operator was not used at all. This result was obtained for both the parameterized and generative representations.

5.2 Generational Model

In this group of design experiments, the impact of the type of the generational model on the fitness of produced design concepts was investigated. Here, two kinds of ES were employed: ES(5,25) with the nonoverlapping generation model and ES(5+25) with the overlapping generational model. All other parameters' values were kept the same.

Figure 9 shows typical results obtained in these experiments. Here, generative representations based on standard CAs were employed for design problem Ia. It clearly shows that ES with the overlapping and nonoverlapping generational models produced almost identical results in terms of both the average best-so-far fitness and the variance of produced results. Essentially the same results were obtained in experiments with parameterized representations.

5.3 Population Sizes

Another group of experiments focused on the determination of the optimal sizes of populations of parents and offspring. Five different combinations of sizes of parent and offspring populations were considered for ES and two combinations for GAs (see Table 3).

Typical results obtained for ES are presented in Figure 10. It shows the results of the ED experiments (problem II) in which three combinations of the parent and offspring population sizes were used, including ES(1+5), ES(5+25), and ES(50+250). Mutation and crossover rates were kept the same in all experiments and equal to 0.025 and 0.2, respectively.

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

Slightly different results were obtained in the experiments with generative representations. Here, four combinations of parent and offspring population sizes were considered: ES(1+5), ES(1+125), ES(5+25), and ES(5+125). Figure 11 shows typical results obtained in these experiments. Specifically, the generative representations based on totalistic CAs were used for problem II. All other EC parameters were kept the same in all experiments whose results are shown in this figure.

Figure 11 clearly shows that the best results were produced with the parent population size equal to 5 and the offspring population size equal to 25. On the other hand, the worst results were produced by the ‘greedy’ ES(1+25). Thus, in the case of generative representations the parallel search conducted by ES(5+25) not only reduced variance of the obtained results, as it was the case with parameterized representations, but is also improved the actual performance of evolutionary processes.

The impact of parent and offspring population sizes was dramatically different for ED processes utilizing GAs. For both combinations of parent and offspring population sizes (see Table 3), i.e., GA(5,25) and GA(50,50), the performance of the algorithms was almost identical. The only difference between the two curves is the reduction of variance for the algorithm with larger population sizes, i.e., for GA(50,50), as it was also observed for ES.

5.4 Evolutionary Algorithm

Design experiments reported in this paper have also investigated the impact of the type of EA on the fitness of produced designs.

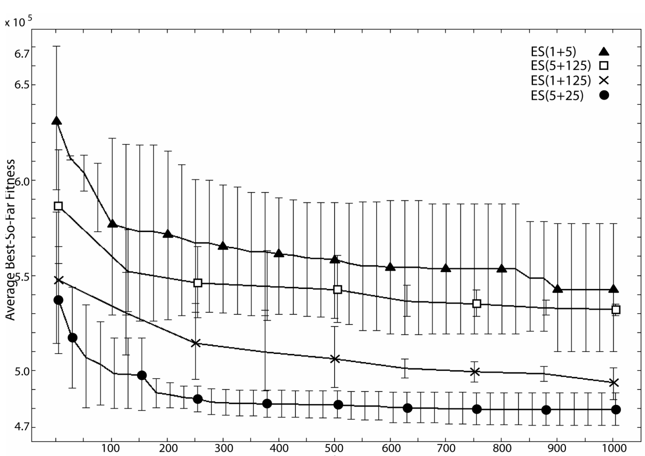

Figure 11. Impact of the populations sizes on the progress of evolutionary processes with generative representations

Figure 12 shows a comparison of the behavior of the two algorithms on problem Ia. Two average best-so-far curves in the upper part of Figure 12 correspond to GAs with two combinations of parent and offspring population sizes, i.e., GA(5,25) and GA(50,50). They are compared to the average best-so-far performance produced by ES with the overlapping ES(5+25) and nonoverlapping ES(5,25) generational model. Optimal mutation and crossover rates, found in previous experiments, were used for ES and GAs. Figure 12 clearly shows that ES outperformed GAs in our problem domain. ES produced designs, on average, 5-8% better than designs produced by GAs.

5.5 Length of Evolutionary Design Processes

In the final group of design experiments, the impact of the length of evolutionary processes on the fitness of produced designs was investigated. In order to do that the results of the long-term evolutionary processes (10,000 evaluations) were compared to the ones obtained in the short-term experiments (1,000 evaluations).

Figure 13 compares the average best-so-far fitness curves obtained in the long-term design experiments (problem II) with generative representations and parameterized representations. EC parameters used in these experiments involved the ‘optimal’ values identified in the short-term processes. This figure shows distinct performance of evolutionary processes utilizing different design representations.

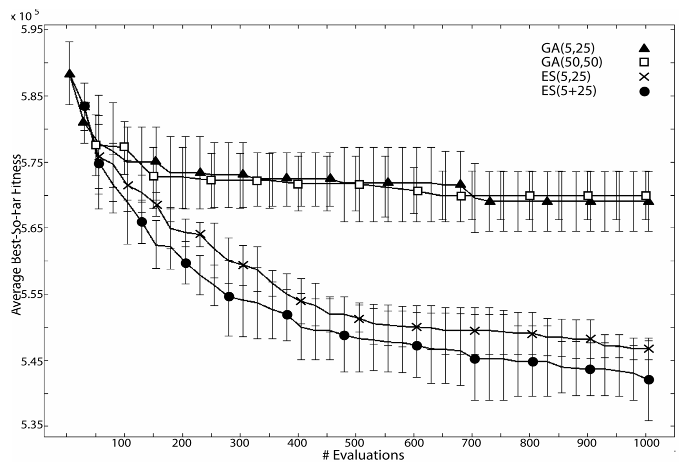

Figure 12. Comparison of the performance of GAs and ES

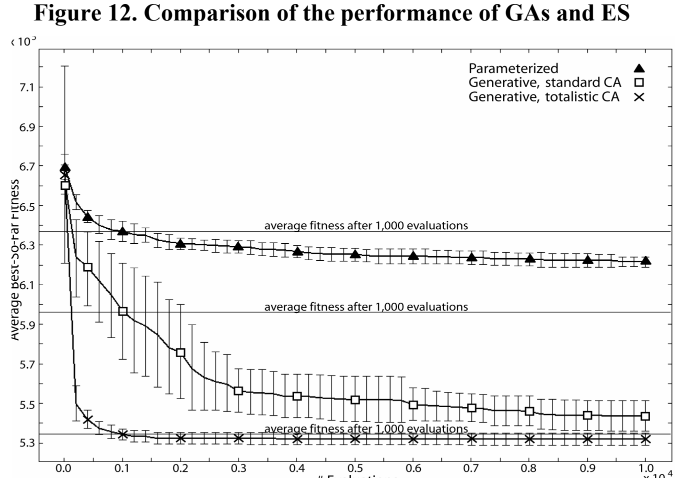

Figure 13. Comparison of the performance of the short-term and the long-term evolutionary design processes

It is clear that generative representations achieved higher average performance improvement in the initial stages of evolution than parameterized representations. In particular, totalistic CAs identified very quickly (in less than 1,000 evaluations) optimal solutions for this problem domain. Similar behavior was observed in the vast majority of conducted design experiments.

6. CONCLUSIONS

The conducted research provided several answers regarding the issues of applicability of parameterized and generative representations in evolutionary design. It also identified the most important EC parameters understood here as the parameters having the biggest impact on the behavior of evolutionary design processes. The 'optimal' values of these parameters were also found for both types of design representations.

First, generative representations (especially those based on totalistic CAs) proved to scale well with the size of the considered design problems. This was, however, not the case with standard CAs in which cells could have a larger set of possible values (problems Ic and II). Here, the genomes were significantly longer than those encoding the parameterized representations.

Second, generative representations significantly outperformed parameterized representations in problem domains where optimal solutions exhibited certain regular (and sometimes hidden) patterns (e.g., problems Ia, Ic, and II). They also identified the optimal regions in these design spaces much faster than parameterized representations.

Third, the design experiments identified the following EC parameters as having the largest impact on the success of evolutionary design processes for both types of design representations: the mutation rate, the size of the population of parents, and the type of an evolutionary algorithm.

The optimal mutation rates for both types of design representations were, however, significantly different. The best evolutionary progress for parameterized representations was achieved for a very low rate of mutation (i.e., 0.025). On the contrary, generative representations produced best results when the high rate of mutation was employed (i.e., 0.3).

Small population sizes were generally preferred by ES. However, too small population sizes increased the variance of the obtained results (both representations) and produced inferior results for generative representations. 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 GAs seemed to be negligible and related only to the variance reduction for the obtained results. It didn't influence the actual performance of the algorithm on these problem domains. The conducted experiments have also shown that ES produced generally superior results than GAs for all design problems considered in this paper.

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, more sophisticated types of generative encodings based on CA, e.g., utilizing non-uniform CAs or encodings with a self-adaptation mechanism, will be investigated and applied to several structural engineering problems.

7. REFERENCES

[1] American Institute of Steel Construction *AISC Manual of Steel Construction*. AISC, Chicago, IL, 1989.

[2] Arciszewski, T., Michalski, R. S. and Wnek, J. Constructive induction: the key to design creativity. In Gero, J. S. and Maher, M. L. eds. *Preprints of the Third International Round-Table Conference on Computational Models of Creative Design*, Heron Island, Australia, 1995, 397-426.

[3] Bentley, P. J. and Kumar, S., Three ways to grow designs: a comparison of embryogenies for an evolutionary design problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, (Orlando, Florida, USA, 1999), Morgan Kaufmann Publishers, 35-43.

[4] Hamda, H., Jouve, F., Lutton, E., Schoenauer, M. and Sebag, M. Compact unstructured representations for evolutionary topological optimum design. *Applied Intelligence*, *16*. 139-155.

[5] Hornby, G. S. *Generative representations for evolutionary design automation*. Ph.D. Dissertation, Brandeis University, Waltham, MA, USA, 2003.

[6] Kicinger, R. *Emergent Engineering Design: Design creativity and optimality inspired by nature*. Ph.D. Dissertation, George Mason University, Fairfax, VA, 2004.

[7] Kicinger, R., Arciszewski, T. and De Jong, K. A. Evolutionary computation and structural design: a survey of the state of the art. *Computers & Structures*. approved.

[8] Kicinger, R., Arciszewski, T. and De Jong, K. A., Morphogenesis and structural design: cellular automata representations of steel structures in tall buildings. In *Proceedings of the Congress on Evolutionary Computation (CEC'2004)*, Portland, Oregon, 2004, IEEE Press, 411-418.

[9] Kicinger, R., Arciszewski, T. and De Jong, K. A. Morphogenic evolutionary design: cellular automata representations in topological structural design. In Parmee, I. C. ed. *Adaptive Computing in Design and Manufacture VI*, Springer-Verlag, London, UK, 2004, 25-38.

[10] Murawski, K., Arciszewski, T. and De Jong, K. A. Evolutionary computation in structural design. *Journal of Engineering with Computers*, *16*. 275-286.

[11] Soh, C. K. and Yang, Y. Genetic programming-based approach for structural optimization. *Journal of Computing in Civil Engineering*, *31*. 31-37.

[12] Wolfram, S. *Cellular automata and complexity: collected papers*. Addison-Wesley, Reading, MA, 1994.