Morphogenesis and Structural Design: Cellular Automata Representations of Steel Structures in Tall Buildings

Rafal Kicinger
CEIE Department
George Mason University
Fairfax, VA 22030 USA
Email: rkicinge@gmu.edu

Tomasz Arciszewski
CEIE Department
George Mason University
Fairfax, VA 22030 USA
Email: tarcisze@gmu.edu

Kenneth De Jong
Computer Science Department
George Mason University
Fairfax, VA 22030 USA
Email: kdejong@gmu.edu

Abstract-**This paper provides the initial results of a study on the applications of generative cellular automata-based representations in evolutionary structural design. First, recent developments in evolutionary design representations and an overview of cellular automata are presented. Next, a complex problem of topological design of steel structural systems in tall buildings is briefly described. Further, morphogenic evolutionary design is introduced and exemplified by cellular automata representations. The paper also reports the initial results of several structural design experiments whose objective was to determine feasibility of the proposed approach. Finally, initial research conclusions are provided.**

I. INTRODUCTION

Evolutionary computation is becoming an increasingly attractive paradigm for civil and structural engineers. It offers a true potential of addressing two important objectives of engineering design. First, it provides structural engineers with a powerful optimization method that can be used to solve many difficult design problems. Second, it is suitable for developing novel/creative designs. This emerging paradigm shift reflects the ongoing transformation of computing in structural engineering from mostly analytical to more holistic aspects of design.

Traditionally, applications of evolutionary methods in structural engineering were focused on structural optimization problems [1]. Currently, together with reported successes of creative evolutionary design [2], we are witnessing an emerging trend of using the evolutionary paradigm to discover novel/creative structural designs. At the same time, complicated models of physical systems are being replaced with distributed models based on simple rules and interactions among elements. It has been shown that even the systems based on very simple rules can produce very complex behavior [3].

Thus, even though the models in structural design are becoming more and more complicated, it is possible that equivalent results can be obtained using very simple rules and programs. Hence, using cellular automata, one of the simplest highly parallel systems based on rules, as generative representations of structural systems seems to be a plausible way of capturing the complex nature of the design process. This approach also offers a potential of developing simple computational models of design processes.

Presently available computing power creates new possibilities for applying this approach to design of engineering systems, even as complex as steel structural systems in tall buildings. The research reported in this paper is the continuation of earlier work reported in [4]. However, the structural design problem discussed here is much more complex than the one investigated previously in which only a wind bracing subsystem was the subject of design and, all other structural members were assumed fixed.

II. BACKGROUND

A. Engineering Design Representations

One of the key issues in evolutionary design, and in evolutionary computation in general, is the choice of an appropriate representation. It becomes even more important when creativity/novelty of generated solutions is one of the research objectives. In the most straightforward evolutionary design applications, each gene represents a dimension of the solution space. In such direct representations each individual consists of a fixed-length string of genes representing some subset of a given set of features. It has been argued that such direct mapping of the problem to its representation is not sufficient when creative/novel solutions are sought [5].

Recently, significant research efforts within the evolutionary design community were focused on studying alternative ways of representing designs. Several researchers investigated indirect and generative representations which do not encode complete design concepts but rather rules which define how to construct these designs [6, 7]. These types of representations are inspired by the processes of morphogenesis occurring in nature which manipulates the rules for growing complex objects, called genetic plans, rather than the complex objects themselves [8].

A representation of an engineering system is usually expressed in terms of attributes which can be either symbolic (when they take values from an unordered or partially ordered set) or numerical (when they take numerical values representing quantities or measurements). Design concepts produced in the conceptual design stage are typically described in terms of symbolic attributes. On the other hand, numerical attributes are used mostly in the detailed design stage [9].

0-7803-8515-2/04/$20.00 ©2004 IEEE

As discussed earlier, applications of evolutionary computation in structural engineering traditionally emphasized optimization side of an evolutionary design process. Hence, in a large majority of applications, structural systems were represented by simple parameterizations consisting of binary, real, or integer-valued attributes [10, 11]. Recently, several interesting encodings of structural systems were proposed including Voronoi-based representations and IFS representations based on fractal theory [12].

B. Cellular Automata

Cellular automata (CAs) are one of the simplest models of highly parallel systems based on local rules. They were initially proposed as models of systems and processes made up of identical, simple, and locally interacting components [13]. Researchers in this field used the simple models to study pattern formation and self-organization processes [14]. It has been discovered that very complex patterns of behavior can be produced out of a set of very simple rules. Recently, it has been suggested that cellular automata and other simple programs may better model nature's most essential mechanisms than traditional mathematical equations [3].

A cellular automaton, contrary to an evolutionary algorithm, is a deterministic system. It is completely defined by giving its initial configuration of cell values and an update rule (called in this paper a CA rule) that transforms a current configuration of cell values to a new one. Each such transformation of the entire configuration of cell values defines one time step. Figure 1 shows how a simple cellular automaton works.

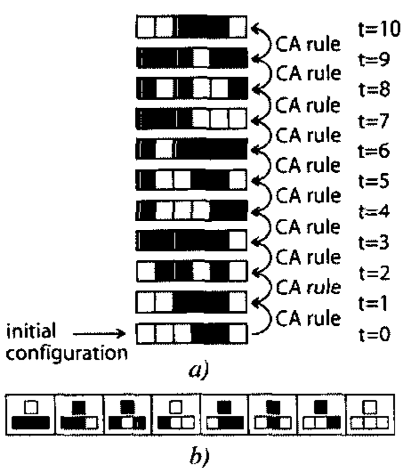


Figure 1. a) Process of iteration of a simple cellular automaton, b) Graphical representation of the CA rule used in part a).

The process of iteration of a simple cellular automaton is shown in Figure 1a). In this particular example, the individual cells can have only binary values, i.e. white squares denote cell values equal to 0 while black squares represents cell values equal to 1. Local neighborhoods affecting the iteration of the currently considered cell are formed by this cell and by its immediate left and right neighbors. Therefore, three cells are considered in each local neighborhood and such neighborhoods are called 'local neighborhoods of size 3'. The bottom row of Figure 1a) consists of 6 squares denoting cells in the initial configuration of a CA. In this particular case, the initial configuration consists of cell state values 0 0 0 1 1 0.

The CA rule that transforms the initial configuration of cells (t=0) into a new one at time t=1, and all subsequent time steps is presented in Figure 1b). It can be interpreted as a complete set of decision rules whose conditions incorporate all possible combinations of cell values in the given local neighborhoods and outcomes determine the values of the currently considered cells at the next time step. Details of the process of transformation of the current configuration of cells at subsequent time steps (*t =1, 2,...*) are illustrated graphically in Figure 2.

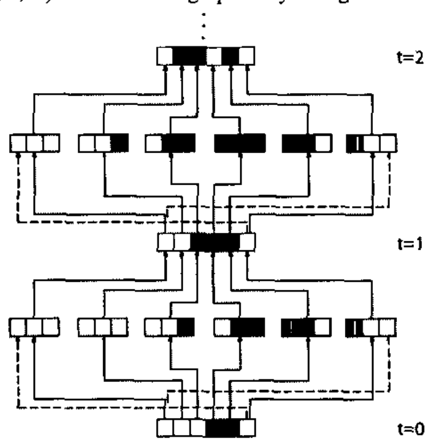


Figure 2. Graphical illustration of the mechanism that determines configurations of cell states at subsequent time steps.

Bottom part of Figure 2 shows the same initial configuration *(t=0)* as in Figure 1a). The process of transforming this initial configuration into a new one at time step *(t=1)* involves several operations. First, a set of local neighborhoods of size 3 is constructed by taking each cell from the initial configuration 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 initial configuration in Figure 2). In this instance, so-called cyclic boundary conditions are used, meaning that the rightmost cell in the initial configuration becomes the left neighbor of the leftmost cell in the initial configuration, and vice versa (denoted by dashed lines in Figure 2). Second, the local neighborhoods created that way are compared to the local neighborhoods which define the conditions of the CA rule (see the bottom row of Figure 1b)). When the two match, the value shown in the top row of Figure 1b) defines the new value of the central cell in the new configuration at the next time step. This process is repeated for each local neighborhood and the obtained values are placed in appropriate positions of the new configuration of cells. In this way, the new configuration is fully defined and at the same one iteration (one time step) of a cellular automaton is completed. The process can be repeated for an arbitrary number of iterations. Figure 1a) shows the results of the iteration process for the first 10 steps. On the other hand, Figure 2 presents a detailed graphical illustration of this process for the first 2 iterations only.

The CA rule shown in Figure 1b) specifies all possible (8 in this case) cell values of a local neighborhood of size three (bottom row) and determines the values achieved by the central cells at the next time step (top row). Thus, if we agree on the ordering of the local neighborhoods as shown in the bottom row of Figure 1b) and assume it fixed, then *any elementary CA rule can be defined by a single eight-digit binary number* specifying the values achieved by the central cells at the next time step for all local neighborhoods.

The number of possible CA rules grows very rapidly with an increase of the number of cell values or the size of the local neighborhood. For example, when the number of possible cell values is equal to 3, and the size of the local neighborhood is the same, there are 7,625,597,484,987 possible CA rules compared to 256 CA rules for elementary CAs. There is, however, a way to significantly reduce the space of possible CA rules by using so-called totalistic CAs. The idea of a totalistic rule is to assume that the new value of the currently considered cell is determined by the average value of this cell and its neighbors, and not on their individual values. For example, a totalistic CA with 3 possible cell values has only 2187 possible CA rules compared to 7,625,597,484,987 rules found in the corresponding standard CA.

CAs have been a subject of significant research efforts in general science, but some initial research has already been conducted in the context of structural design. Self-organizations of topologies in mechanical structures was studied in Inou et al. [15]. Kundu et al. [16] applied CAs to shape optimization of structural plates. Kita and Toyoda used CAs to shape and topology optimization of two-dimensional elastic structures [17] and sizing optimization of truss structures [18]. Hajela and Kim [19] applied genetic algorithms to search the space of CA rules in structural analysis of 2D elastic structures.

C. Steel Structures in Tall Buildings

Steel skeleton structures in tall buildings are considered the most complicated structures designed and built, comparable in their conceptual and physical complexity only with large span bridges. Usually, such structures are 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 columns and beams to increase the flexural rigidity of the entire system and that is driven mostly by stiffness requirements related to wind forces.

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, including gravity, wind and earthquake loads, deformations, vibrations, 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 an optimal configuration of structural members, called a "design concept." It is difficult, complex, and still not fully understood domain of structural engineering.

III. Morphogenesis and Structural Design

Parameterized representations of engineering systems have been predominant in the applications of evolutionary methods to structural design. They proved to perform well when solutions to strictly optimization problems were sought. They are, however, not sufficient when the issues of inventive/creative design become equally important to the optimality of produced design concepts. Several types of types of generative representations have been proposed to remedy this problem. In the previous work [4], the authors proposed generative representations of wind bracing systems in tall buildings based on one-dimensional and two-dimensional cellular automata. These representations were inspired by the process of the embryological development of the structure of an organism occurring in nature and called morphogenesis.

A. Morphogenic Evolutionary Design

In the morphogenic evolutionary design introduced in [4], a structural design concept is produced from a 'design embryo' using a 'design rule' which is applied to the design embryo to build from it the structure of a wind bracing system. Figure 3 provides a simple example of this approach in which a design concept of a wind bracing subsystem is constructed from a generative representation encoding a design embryo (leftmost genes a-f in Figure 3a)) and a design rule (rightmost genes 1-8 in Figure 3a)).

This representation has been developed using the concept of division of the structural grid of a tall building (the system of vertical and horizontal lines of columns and beams, respectively) into cells. A cell is understood here as a part of the structural grid contained within the adjacent vertical and horizontal grid lines. Design embryo is formed by a one-dimensional lattice of cells representing the initial configuration of cell values and at the same time determines the configuration of the first story in a wind bracing system of a tall building (see the configuration at $t=0$ in Figure 3c)). Design rule is simulated by a one-dimensional cellular automaton (1D CA) rule (see Figure 3b)). It consists of a complete set of decision rules whose conditions (bottom part of Figure 3b)) incorporate all possible combinations of cell values (in this example representing types of bracings) in the given local neighborhoods and outcomes specifying the values of the central cells of these neighborhoods at the next time step (top part of Figure 3b)). The length of the design embryo is equal to the number of bays in a tall building (see Figure 3c)). On the other hand, the length of the design rule simulated by a 1D CA depends on the number of possible cell values and the size of the local neighborhood.

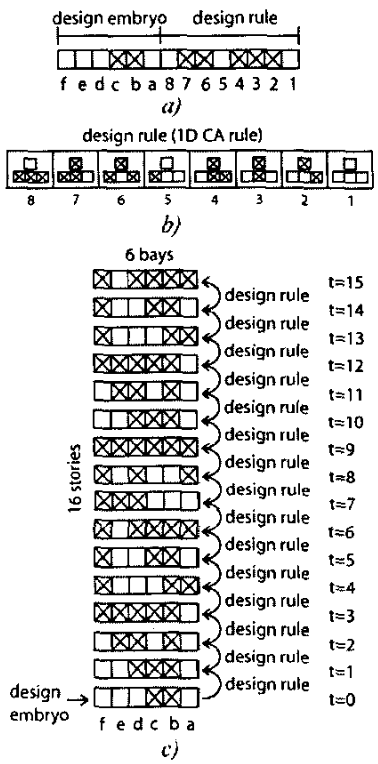


Figure 3. Process of constructing a design concept of a wind bracing system in a tall building from a design embryo and using a design rule applied to this embryo.

The example of a design rule simulated by a 1D CA shown in Figure 3b) is simple and based on only two possible cell values (representing no bracing and X bracing), and neighborhood of size 3. The process of building a design concept of a wind bracing subsystem from the design embryo and using the design rule is shown in Figure 3c). The design rule is iterated for the number of steps that is one less than the number of stories in a tall building and thus forms a design concept which is subsequently evaluated.

Bottom part of Figure 3b) shows all possible combinations of conditions for this design rule. In this particular example, they are ordered from 1 to 8. If this ordering is assumed the same for the entire class of design rules with binary cell state values and local neighborhood of size 3, then the outcome values (shown in the top part of Figure 3b)) uniquely define every rule belonging to this class. This fact has been used in the definition of the structure of the genome shown in Figure 3a). Here, genes 1-8 encode the outcome values produced by the design rule presented in Figure 3b) and, given the assumed ordering, uniquely define it.

B. Generative Representations of Steel Structures

Generative representations of steel structural systems investigated in [4] and briefly discussed in the previous section were focused only on one, albeit important, part of the system, i.e., on a subsystem of wind bracings in a tall building. A complete design concept of a steel structural system, however, should contain not only wind bracings, but also beams, columns, and supports.

An approach to encode complete design concepts of steel structures in tall buildings is proposed in this section. It utilizes the idea of combining in one genome several generative representations of various subsystems of a steel structure. To achieve it, an approach similar to the one described in the previous section is employed. Figure 4 shows the schematic view of the structure of a linear genome representing a complete design concept.

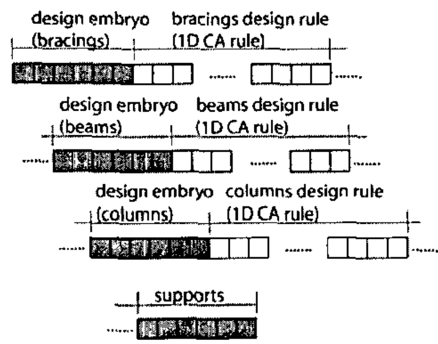


Figure 4. A schematic view of a generative representation of a complete design concept of a steel structural system in a tall building.

The genome encodes design embryos of wind bracing subsystem, beam subsystem, and column subsystem (gray cells) and design rules simulated by 1D CA rules (white cells). The design rules are applied to their corresponding design embryos and used to build from them the individual subsystems. A configuration of supports in a tall building is encoded at the end of the linear genome (gray cells). The support configuration, however, is not iterated in this particular case.

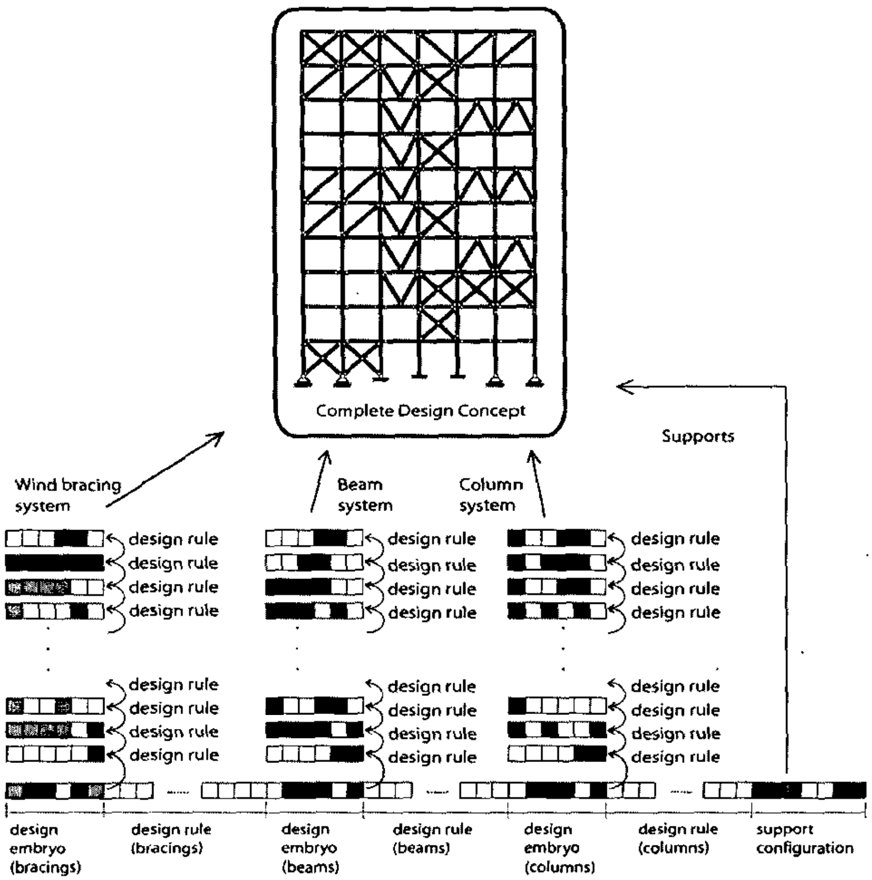


Figure 5. Process of constructing a complete design concept of a steel structural system in a tall building from the generative representation.

Each design rule is applied to its corresponding design embryo and iterated the number of times that is one less than the number of stories in a tall building. In this way, systems of wind bracing, beams, and columns are formed. Once the complete configurations of all subsystems of a steel structure are constructed, they are assembled together to form a complete representation of a design concept, which is subsequently evaluated. The process of constructing a complete design concept from this representation is presented in Figure 5.

Significant difference of this representation compared to the generative representation of a wind bracing subsystem discussed in the previous section is that the genome is no longer homogenous. Various parts of the genome encode different subsystems of the steel structure and hence different attributes are used to represent them. These attributes, in general, can have different number of possible values.

The advantages of this approach are similar to the ones described in [4], namely compactness and excellent scalability. For example, when a genome encoding a complete design concept of a structural system with 30 stories and 6 bays and consisting of a wind bracing system with 7 types of bracings, a beam system with 2 types of beams, a column system with 2 types of columns, and with 2 types of supports is considered, it has 365 genes. This corresponds to 576 genes that have to be used in standard parameterized representations. In the case when the design rules are simulated by the totalistic 1D CA rules, the genome is even more compact and its length is reduced to only 33 genes.

The disadvantages of this representation include the lack of diversification of the design rules. Each subsystem in a steel structure is designed using a single design rule which is applied at each story. Hence, it is impossible to diversify design rules for various parts of the subsystem, e.g. in traditional design different design rules may be used in the bottom part of the structure, where internal forces are the largest, compared to the upper part of the structure where internal forces are the smallest but local stiffness requirements are the same. Additional drawback includes a necessity to create a specialized mutation operator that would manipulate non-homogeneous genomes. The modifications required in adapting a standard mutation operator to this representation should be minimal, though.

IV. Experimental Results

Initial experiments reported in this paper were aimed to determine the feasibility of morphogenic evolutionary design of complex structural systems. This objective has been realized through the analysis of the results of a number of experiments in which the generative representation proposed in the previous section was used and through the comparison with the results produced using the parameterized representations similar to those used in [20]. Design experiments were conducted using an experimental research and design tool, called *Emergent Designer*, developed at George Mason University [4]. It is a Java-based system intended for both the design experiments in the area structural design as well as the analysis of the design processes from the perspective of complex adaptive systems and dynamical systems.

A. Experimental Design

Table 1 shows assumed parameters and their values used in the experiments. The subject of design were steel structures of a 30-story building with 6 bays. The following subsystems of steel structures were evolved: wind bracing subsystem, beam subsystem, and supports. Column subsystem was not evolved and all column types in steel structures were assumed the same during the entire design process.

Seven types of wind bracing members were considered in the design of a wind bracing subsystem. Their phenotypic, symbolic, and genotypic representations are shown in Figure 6. Beam subsystem was formed using two types of beam members presented in Figure 7a)-c). Genes representing supports had two possible values as it is shown in Figure 7d)-f).

Two types of generative representations of steel structural systems were experimentally investigated. In the first type, encoded design rules were simulated by standard 1D CA rules and the length of the genome was equal to 370. In the second type, totalistic 1D CA rules were used and, in this case, the genome consisted of 42 genes. The results obtained using both types of generative representations were subsequently compared to the results produced using parameterized integer-valued representations, similar to those used in [20]. Parameterized representations were 367 genes long. As mentioned before, the genomes were non-homogeneous for all types of representations discussed above.

Evolution strategies (ES) [21] were used to evolve representations of steel structures in tall buildings. Four combinations of parent and offspring population sizes (involving either 1 or 5 parents and 25 or 125 offspring) were studied for all 3 types of representations. All other EA parameters were held constant as shown in Table 1.

TABLE 1. Experimental Parameters and Their Values

Parameter	Value(s)
Domain parameters:	
Number of bays	6
Number of stories	30
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	first order
Beams	pinned, fixed
Column	fixed
Supports	pinned, fixed
Wind bracings	no bracing, diagonal bracing (/), diagonal bracing (\), K bracing, V bracing, simple X bracing, and X bracing
CA representation parameters:	
CA type	1D, 1D totalistic
CA neighborhood radius	1
Number of CA cell values	7 (bracings), 2 (beams)
Evolutionary algorithm parameters:	
EA	ES
Pop. sizes (parent, offspring)	(1,25), (5,25), (1,125), (5,125)
Generational model	overlapping
Selection (parent, survival)	(uniform stoch., truncation)
Mutation rate	1/L (L- length of genome)
Crossover (type, rate)	(uniform, 0.2)
Fitness	weight of the steel structure (minimization problem)
Initialization method	random
Constraint handling method	death penalty (infeasible designs assigned 0 fitness)
Simulation parameters:	
Termination criterion	1,000 fitness evaluations
Number of runs	5 (in each experiment)

The fitness of individual designs was determined by the total weight of the steel structure. It was calculated using a structural analysis, design and optimization package called SODA, developed by Waterloo Systems in Waterloo, Ontario, Canada, which forms one of the components of *Emergent Designer*. The optimization (minimization) of weight of steel skeleton structures was conducted in two stages. In the first stage, evolutionary algorithm optimized the topology of steel structural systems in tall buildings as discussed earlier. The second stage of optimization (sizing optimization) was conducted by SODA. Here, cross-sections of all structural members (including beams, columns, and wind bracings) were optimized with respect to the total weight of the steel structure.

Figure 6. a) Phenotypic representation of the evolved wind bracing members, b) Their symbolic values, c) Genotypic values corresponding to appropriate wind bracing types.

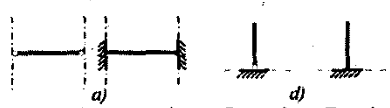


Figure 7. a)-c) Phenotypic, symbolic, and genotypic representation of the evolved beam members, d)-f) Phenotypic, symbolic, and genotypic representation of the evolved supports.

All design experiments reported in this paper consisted of 5 runs, each started with a different random seed value. Each run involved a fixed budget of 1,000 fitness evaluations.

B. Initial Design Experiments

The conducted experiments have shown that the best results in terms of fitness were obtained when the parent population size consisted of more than one individual, no matter what type of representation was used. The overall best results were produced by the totalistic 1D CA representations with parent and offspring population sizes equal to 5 and 25, respectively. Figure 8 shows the average best-so-far fitness obtained in these experiments for all 3 types of representations (vertical bars denote 95% confidence intervals calculated using Johnson's modified *t* test). It is clearly visible, and at the same time statistically significant, that totalistic 1D CA representations outperformed the other two types of representations which is in concordance with authors' previous findings reported in [4].

Parameterized representations produced the best results when population sizes of 5 and 125 were used. Even in this case, totalistic 1D CA representations generated better results, as it is shown in Figure 9. Figure 10a) shows the best design produced in the reported experiments. It was generated using the totalistic 1D CA representation. Its fitness (the total weight of the structural system) is about 13% better than for the best design produced using parameterized representations. It is a significant improvement rarely possible when using parameterized representations. The produced design strongly resembles a traditional structural system in the form of the rigid braced frame with the exception of the ground floor bracings. In this design, a wind bracing subsystem is incrementally developed, beginning at the ground level, using a design rule which places K bracings in all bays while at the same time another design rule constructs a beam subsystem using exclusively fixed beams.

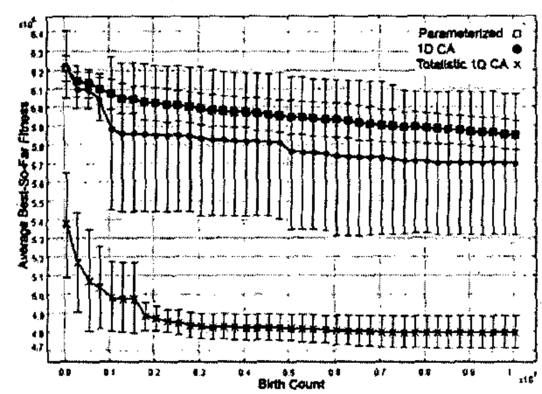


Figure 8. Average best-so-far fitness for 1D CA, totalistic 1D CA, and parameterized representations with parent and offspring population sizes equal to 5 and 25, respectively.

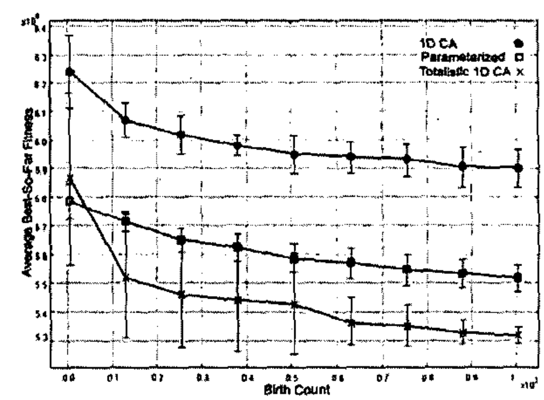
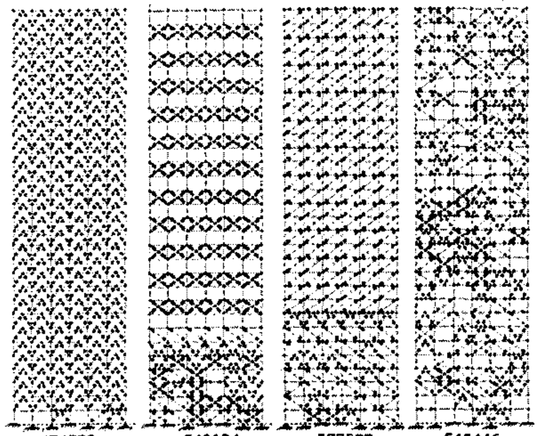


Figure 9. Average best-so-far fitness for 1D CA, totalistic 1D CA, and parameterized representations with parent and offspring population sizes equal to 5 and 125, respectively.

The experiments produced several interesting structural shaping patterns, found using both totalistic 1D CA representations (see Figure 10b)) and standard 1D CA representations (see Figure 10c)). These structural shaping patterns are qualitatively different than the ones generated by parameterized representations (see Figure 10d)). Surprisingly, the design concept shown in Figure 10b) is similar to the known concept of a rigid frame with several horizontal trusses situated at various levels. The horizontal trusses redistribute the wind forces and subsequently improve the performance of the structural system.



474780 a) 540124 b) 577577 c) 545446 d)

Figure 10. a) Best design produced in the reported experiments, b-c) Examples of interesting structural shaping patterns, d) Best design produced by the parameterized representation. The fitness values (in lbs.) are reported below each design.

V. CONCLUSIONS

Research reported in this paper is the continuation of the previous work on the morphogenic evolutionary design. Here, the preliminary results are provided concerning the use of this new design paradigm applied to much more complex structural design problems than previously reported. The initial findings are encouraging and confirm our previous conclusions regarding the feasibility of cellular automata-based generative representations in structural design. These representations proved to perform well both in generating optimal design concepts as well as in producing interesting structural shaping patterns.

The advantages of generative representations with respect to parameterized representations can be explained in computational terms. A more intriguing question, and an obvious objective of further research, is if these advantages are also caused by the fact that human designers usually create design concepts using various heuristics (design rules) which are gradually applied to the individual parts of a building, and this process is, at least partially, mimicked by the use of cellular automata.

REFERENCES

[1] R. Kicinger, T. Arciszewski, and K. A. De Jong, "Evolutionary computation and structural design: a survey of the state of the art," *Computers & Structures*, submitted, 2004.

[2] P. J. Bentley and D. W. Corne, "Creative evolutionary systems." San Francisco, CA: Morgan Kaufmann Publishers, 2002.

[3] S. Wolfram, *A new kind of science*. Champaign, IL: Wolfram Media, 2002.

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

[5] G. S. Hornby, "Generative representations for evolutionary design automation," in *Department of Computer Science*. Waltham, MA: Brandeis University, 2003.

[6] G. S. Hornby and J. B. Pollack, "Creating high-level components with a generative representation for body-brain evolution," *Artificial Life*, vol. 8, 2002.

[7] P. J. Bentley and S. Kumar, "Three ways to grow designs: a comparison of embryogenies for an evolutionary design problem," Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99), Orlando, Florida, USA, 1999.

[8] D. A. W. Thompson, *On growth and form*. Cambridge: Cambridge University Press, 1942.

[9] T. Arciszewski, R. S. Michalski, and J. Wnek, "Constructive induction: the key to design creativity," in *Preprints of the Third International Round-Table Conference on Computational Models of Creative Design*. Heron Island, Queensland, Australia, 1995, pp. 397-426.

[10] P. Hajela and E. Lee, "Genetic algorithms in truss topological optimization," *Journal of Solids and Structures*, vol. 32, pp. 3341-3357, 1995.

[11] C. D. Chapman and M. J. Jakiela, "Genetic algorithm-based structural topology design with compliance and topology simplification considerations," *Journal of Mechanical Design*, vol. 118, pp. 89-98, 1996.

[12] H. Hamda, F. Jouve, E. Lutton, M. Schoenauer, and M. Sebag, "Compact unstructured representations for evolutionary topological optimum design," *Applied Intelligence*, vol. 16, pp. 139-155, 2002.

[13] J. von Neumann, "The general and logical theory of automata," in *John von Neumann, Collected works*, vol. 5, A. H. Taub, Ed., 1963, pp. 288.

[14] S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of Modern Physics*, vol. 55, pp. 601-644, 1983.

[15] N. Inou, T. Uesugi, A. Iwasaki, and S. Ujihashi, "Self-organization of mechanical structure by cellular automata," in *Fracture and strength of solids. Part 2: Behavior of materials and structure*, P. Tong, T. Y. Zhang, and J. Kim, Eds., 1998, pp. 1115-1120.

[16] S. Kundu, J. Oda, and T. Koishi, "A self-organizing approach to optimization of structural plates using cellular automata," Proceedings of the Second World Congress on Structural and Multidisciplinary Optimization (WCSMO-2), Zakopane, Poland, 1997.

[17] E. Kita and T. Toyoda, "Structural design using cellular automata," *Structural and Multidisciplinary Optimization*, vol. 19, pp. 64-73, 2000.

[18] E. Kita and T. Toyoda, "Truss structural design based on concept of cellular automata," Proceedings of the International Association for Shell and Spatial Structures Symposium (IASS2001), Nagoya, Japan, 2001.

[19] P. Hajela and B. Kim, "On the use of energy minimization for CA based analysis in elasticity," *Structural and Multidisciplinary Optimization*, vol. 23, pp. 24-33, 2001.

[20] K. Murawski, T. Arciszewski, and K. A. De Jong, "Evolutionary computation in structural design," *Journal of Engineering with Computers*, vol. 16, pp. 275-286, 2001.

[21] I. Rechenberg, *Evolutionsstrategie; Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart-Bad Cannstatt: Frommann-Holzboog, 1973.