GENERATIVE REPRESENTATIONS IN STRUCTURAL ENGINEERING

Rafal Kicinger[1], Tomasz Arciszewski[2], and Kenneth De Jong[3]

ABSTRACT

This paper proposes a new approach to representing structural system inspired by various models of complex systems. Several types of generative representations of steel structural systems are provided and empirically investigated. These representations utilize various kinds of cellular automata to generate design concepts of steel structures in tall buildings. In the paper, a brief overview of the state-of-the-art in cellular automata and generative design is presented. Next, several types of generative representations of steel structural systems in tall buildings are described. The paper also reports the results of several design experiments. They have shown that generative representations produce novel structural shaping patterns which are qualitatively different than the patterns obtained using traditionally used parameterized representations. They also significantly improve the performance of evolutionary algorithms optimizing the structural systems. Finally, research conclusions are presented and most promising paths of future research are discussed.

KEY WORDS

Generative representations, morphogenic evolutionary design, evolutionary computation, structural design.

INTRODUCTION

Computational approaches to structural conceptual design require the formulation and use of a formal design representation space in which design concepts are sought. Therefore, building such space is extremely important and can be compared to a knowledge acquisition process (Arciszewski et al. 1995). Specifically, in this process background design knowledge is transformed into a system of nominal attributes describing structural system's function, form, requirements, constraints, etc. The values of these attributes potentially describe not only all known design concepts but also still unknown yet feasible concepts. For this reason, the extent and nature of a representation space is crucial when novelty is sought. Thus, a designer's goal strongly influences the choice of a particular representation.

[1] Postdoctoral Fellow, Civil, Environmental and Infrastructure Engineering Department and Critical Infrastructure Protection Program, George Mason University, 4400 University Drive, MS 4A6, Fairfax, VA 22030, Phone 703-993-1658, Fax 703-993-1521, rkicinge@gmu.edu

[2] Professor, Civil, Environmental and Infrastructure Engineering Department, George Mason University, 4400 University Drive, MS 4A6, Fairfax, VA 22030, Phone 703-993-1513, Fax 703-993-1521, tarcisze@gmu.edu

[3] Professor, Computer Science Department, George Mason University, 4400 University Drive, MS 4A5, Fairfax, VA 22030, Phone 703-993-1553, Fax 703-993-1710, kdejong@gmu.edu

When the focus is on finding optimal designs, the designer's attention is usually limited to a specific design concept or to several concepts of known designs. In this case, design representations usually take a form of *parameterizations* of engineering systems, or their parts. In evolutionary design (ED) applications, these attributes are then 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 1c). In this case, symbolic attributes (see Figure 1b) are directly encoded and they represent types of structural members (see Figure 1a). In other words, in a parameterized representation, there is a one-to-one mapping between the attributes describing the structural system and the genes in a genome.

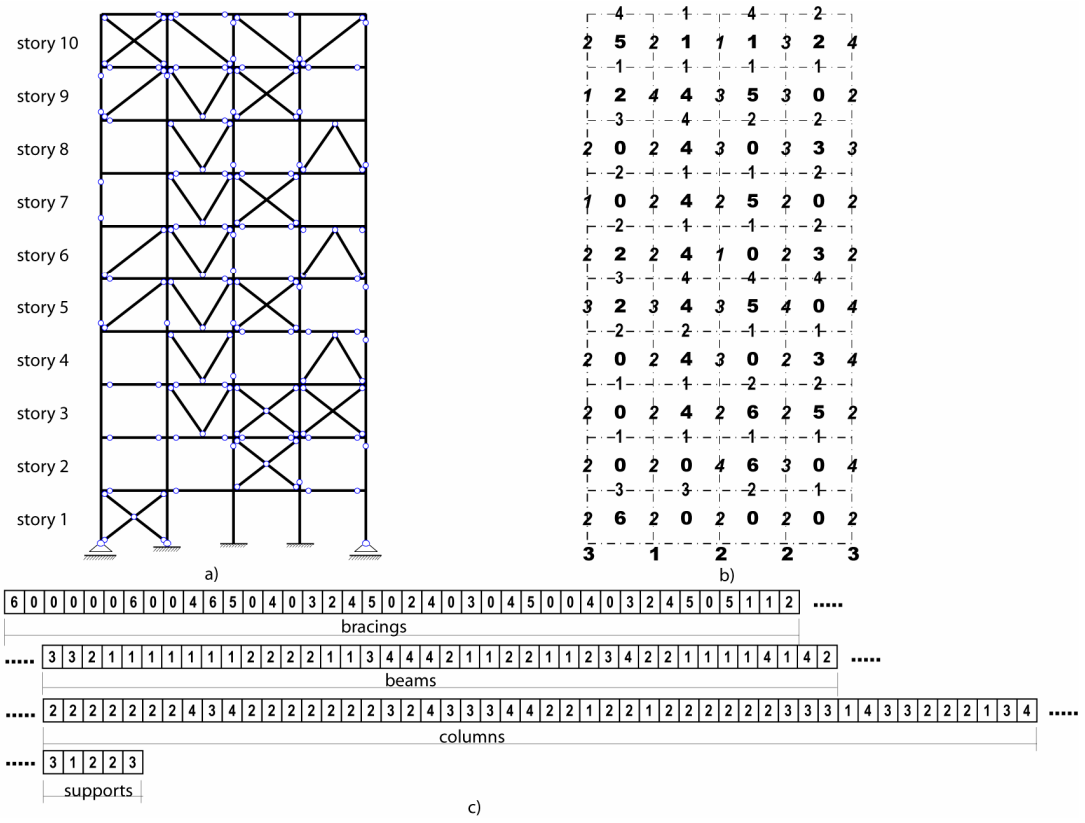

Figure 1: Example of a parameterized representation of a steel structural system

The situation is different when the designer wants to find novel designs. In this case, the parameterized representations are insufficient and more general and usually more complex representations should be employed.

In the paper we specifically address this problem. We propose and experimentally investigate several types of generative representations of structural systems based on cellular automata (CAs). We show that these representations not only outperform traditionally used parameterized representations in optimizing structural systems in tall buildings, but they also produce novel and qualitatively different structural shaping patterns.

BACKGROUND

CELLULAR AUTOMATA

Cellular automata are one of the simplest models of complex systems. They can be used to model complex systems and processes consisting of a large number of *identical*, *simple*, and *locally interacting* components. They have been successfully applied in many disciplines of science, including physics, biology, chemistry, economy, geology, etc. In structural engineering, they have been used, for example, in shape optimization (Inou et al. 1994), and in optimization of cross-sections in truss structures (Kita and Toyoda 2001).

CAs can also be thought of as useful idealizations of the dynamical behavior of various systems. They capture many essential features of complex *self-organizing* cooperative behavior observed in real world systems. Hence, they have been used to study *pattern formation* and *self-organization* processes. Their ability to exhibit very complex patterns of behavior out of a set of relatively simple rules has generated great interest over the last forty years. Recently, Wolfram (2002) suggested that cellular automata and other simple programs may better model nature's most essential mechanisms than traditional mathematical equations.

CAs are fully deterministic systems. They are uniquely defined by their transformation rule (called in this paper a CA rule) and an initial configuration of cell values. A CA rule is 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 considered cells (usually central cells in a local neighborhood) at the next time step. The CA rule is applied to the current configuration of cell values and the result of this operation uniquely determines the configuration of cell values at the next time step. This process is illustrated graphically in Figure 2. Specifically, it shows the simplest possible CA, called an elementary CA, which consists of a one-dimensional lattice of cells and in which each cell can only be in one of two possible states (illustrated here by white and black squares). Moreover, the value of each cell at a next time step is determined by a transformation rule (illustrated graphically in Figure 2c) which considers only values of 3 cells located in a local neighborhood of a given cell: the value of the cell itself and the values of its two immediate neighbors.

Figure 2b shows how elementary CAs work. The bottom row of Figure 2b consists of 6 squares (cells) denoting an initial configuration of cells at time $t=0$. In this particular case, the initial configuration consists of cell values 0 0 0 1 1 0. The specific CA rule used to iterate this initial configuration for 2 time steps is presented in Figure 2b.

First, a set of local neighborhoods of size 3 (i.e. composed of 3 cells) 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 2b). In this particular example, 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 2b). Next, the local neighborhoods created that way are compared to the local neighborhoods shown in the bottom row of Figure 2c (conditions of the transformation rule).

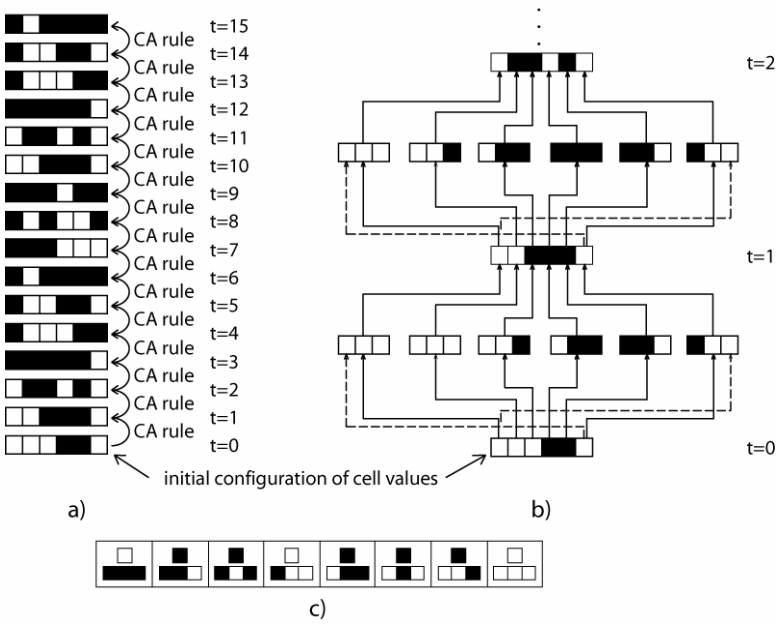

Figure 2: Process of iteration of an elementary cellular automaton

When the two match, the value shown in the top row of Figure 2c (output of the transformation rule) defines the new value of the central cell at the next time step. This process is repeated for each local neighborhood and the values obtained are placed in appropriate slots in the new configuration of cells at time *t=1* thus completely defining this configuration. The process is repeated for an arbitrary number of steps. Figure 2a shows the results of the iteration process for the first 15 steps. Figure 2b gives a detailed representation of the process of determining the new configuration at a subsequent time step for the first 2 iterations only.

The simplest possible CAs shown in Figure 2 can be generalized in many ways. One of them is increasing the number of cell values. This, however, expands the number of possible CA rules and hence makes the CA rule space larger. For instance, when the number of cell values increases to 3 then there are 7,625,597,484,987 possible CA rules compared 256 possible CA rules for CAs with binary cell values (with the same size of the local neighborhood equal to 3). Increasing the number of cell values even further causes a rapid growth of the size of the CA rule space. There is, however, a way to significantly reduce it by introducing a concept of a totalistic CA. A totalistic CA is based on the assumption that the value of each cell at a next time step depends only on the *average* value of the cell itself and the neighboring cells, and not on their individual values (Wolfram 2002). For example, there are only 2187 possible totalistic CA rules with 3 cell values compared to 7,625,597,484,987 rules found in the corresponding standard CAs.

Another possible generalization extends the number of dimensions of the lattice of cells. Thus, two-dimensional cellular automata (2D CAs) are generalizations of one-dimensional systems in which the lattice of cells has two dimensions. In order to completely define 2D CAs we can use the same set of parameters used for 1D CAs, but with one additional property. Namely, in order to fully define a 2D CA transformation rule (a 2D CA rule) one

not only has to specify the radius of the local neighborhood (the number of neighboring cells included in the local neighborhood), but also its shape. Figure 3 shows the shapes of the local neighborhoods investigated in the design experiments reported in this paper.

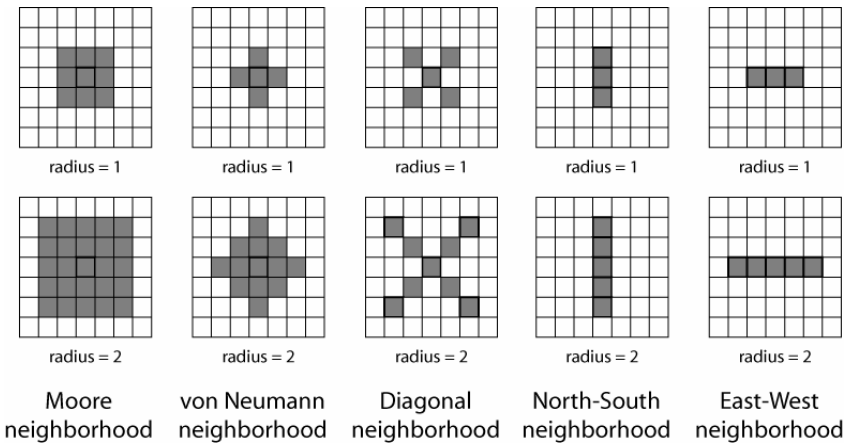

Figure 3: Shapes of the local neighborhoods in 2D CAs

DESIGN REPRESENTATIONS

Representations of engineering designs are one of the key elements in any evolutionary design application and have been recently a subject of significant research investigations (Bentley and Kumar 1999; Hornby 2003). The vast majority of applications of evolutionary methods in structural engineering, with few notable exceptions, used simple parameterizations of engineering systems (see Figure 1) encoded in terms of binary, real, or integer-valued attributes. These representations performed well when the emphasis was on optimization of engineering designs, particularly during the detailed design stages. They were, however, not sufficient when both novelty of generated designs and their optimality were regarded as equally important objectives.

Recently, several researchers proposed and studied *indirect* and *generative* representations inspired by the processes of *morphogenesis* occurring in nature (Bentley and Kumar 1999). These representations, contrary to the parameterized representations discussed earlier, do not encode complete design concepts but rather rules on how to develop, or 'grow,' these designs. It has been shown that generative representations improve the scalability of evolutionary design systems (Hornby 2003) and produce novel design concepts exhibiting interesting and qualitatively different patterns (Kicinger 2004). The state-of-the-art in ED can be found in (Arciszewski and De Jong 2001; Kicinger et al. 2005b).

GENERATIVE REPRESENTATIONS OF STEEL STRUCTURAL SYSTEMS

In this section we define several types of generative representations of steel structural systems in tall buildings based on one-, and two-dimensional CAs. First, generative representations of a wind bracing system in a tall building are proposed. Here, both 1D CAs and 2D CAs are employed. Next, more complex generative representations of entire steel structural systems in tall buildings are defined.

WIND BRACING SYSTEMS IN TALL BUILDINGS

In this section, generative representations of wind bracing systems in tall buildings are presented. In this case, only a subsystem of wind bracings in a steel structural system is considered while the configurations of the remaining subsystems i.e. beam subsystem, column subsystem, and supports, are assumed the same during the entire design process.

One-Dimensional Cellular Automata

This type of generative representation uses a single one-dimensional initial configuration of cells, called here a 'design embryo,' and a single 1D CA rule, called here a 'design rule,' to construct a design concept of a wind bracing system (see Figure 4a). A cell is understood here as a part of the structural grid contained within the adjacent vertical and horizontal grid lines (the system of vertical and horizontal lines of columns and beams, respectively).

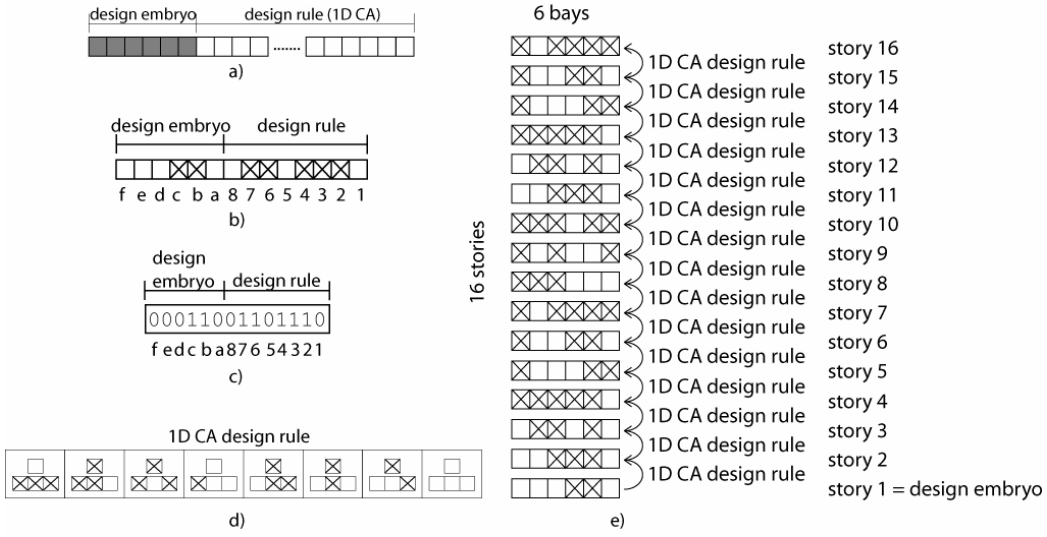

Figure 4: Generative representation of a wind bracing system based on 1D CAs

The design embryo represents the initial configuration of cell values from which a design concept is developed and at the same time it forms the configuration of the first story in a wind bracing system of a tall building (see the configuration of story 1 in Figure 4e). The design embryo is encoded in the first part of the genome (see genes a-f in Figure 4b) whereas the second part encodes the design rule (see genes 1-8 in Figure 4b). The representation of the design rule contains the output values produced by a 1D CA rule (see the top row of Figure 4d) which uniquely define any 1D CA rule (assuming the same ordering of all possible combinations of cell values (types of bracings) in the given local neighborhoods shown in the bottom row of Figure 4d).

In this type of the generative representation, the length of the design embryo is equal to the number of bays in a tall building while the length of the encoding of the design rule depends on two factors: the number of possible cell values and the size of the local neighborhood. Figure 4e illustrates the process of developing, or 'growing,' a design concept of a wind bracing system from the design embryo and using the design rule shown in Figure

4d. The design rule is iterated for the number of steps that is one less than the number of stories in a tall building. The result of this developmental process is a fully defined design concept of a wind bracing system which is subsequently evaluated.

Figure 4 shows a simple example of a wind bracing system composed of only two types of wind bracing elements (X bracings and no bracings (empty cells)) and the local neighborhood of size 3. Such design concepts are represented by binary strings (see Figure 4c) which are easily manipulated by evolutionary algorithms. In this case, the total length of the genome encoding a wind bracing system with 6 bays and 16 stories (see Figure 4e) is equal to 14. On the contrary, 96 genes would need to be employed to represent the same design using the parameterized representation. Thus, the advantages of this type of generative representation include compactness and excellent scalability.

Two-Dimensional Cellular Automata

Another type of generative representation of a wind bracing system is based on 2D CAs. It is essentially composed of the same elements, i.e. the design embryo and the design rule (see Figure 5d). The design embryo, however, has now a form of a 2D array which represents the configuration of the entire wind bracing system (see Figure 5d). Also, the design rule is now based on 2D CA rule (see Figure 5c). The process of developing a design concept using this representation is presented in Figure 5d. As before, the design rule is applied to the design embryo and iterated a number of times. The number of iterations is called the *iteration_max* parameter.

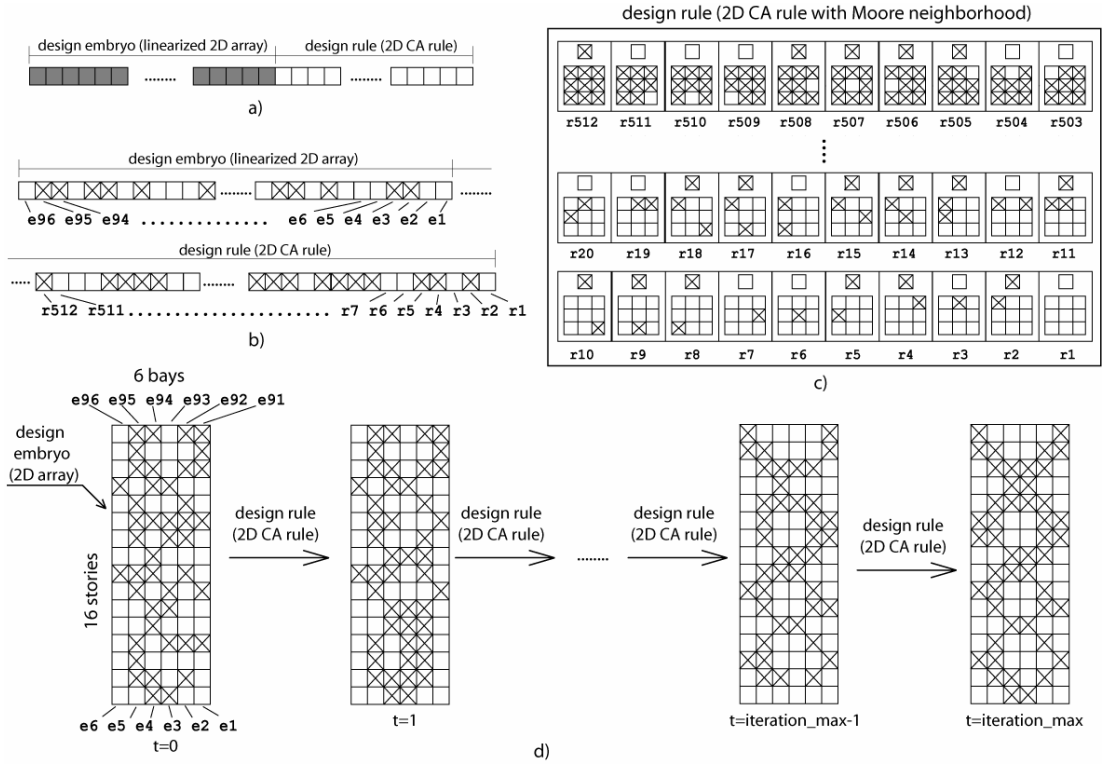

Figure 5: Generative representation of a wind bracing system based on 2D CAs

The final configuration obtained during this developmental process forms a design concept which is subsequently evaluated. The advantages of this representation include the possibility of explicit modeling of local interactions among structural members in a wind bracing system. This is easily achieved by varying the shape of the local neighborhood and its radius (see Figure 3). This, however, comes at a cost of significantly reduced scalability when compared to the representations based on 1D CAs (the genomes are significantly longer). Increasing the number of values in a 2D CA (types of wind bracing elements) or the size of the local neighborhood (range of interactions) causes a rapid growth in complexity. Hence, only totalistic 2D CAs (which significantly reduce the size of the CA rule spaces) can be used for most purposes.

ENTIRE STEEL STRUCTURAL SYSTEMS IN TALL BUILDINGS

In the previous section, generative representations of a wind bracing system in a tall building were presented. They are extended here to represent the entire steel structural system, i.e. wind bracings, beams, columns, and supports. An approach used in this case is based on combining several generative representations of various subsystems of a steel structure in a single genome (see Figure 6a). The subsystems are developed from the initial design embryos by applying the corresponding design rules in an iterative process described earlier.

Figure 6b shows the process of constructing various subsystems from one-dimensional design embryos and the design rules based on 1D CA rules. When all subsystems are fully developed, they completely define the configurations of all members of the steel structural system which is subsequently evaluated. The advantages of this representation are similar to the ones exhibited by the generative representations of wind bracing systems based on 1D CAs, i.e. compactness and excellent scalability.

EXPERIMENTAL RESULTS

Design experiments were conducted using an experimental research and design support tool, called *Emergent Designer*, developed at George Mason University (Kicinger et al. 2005a). Table 1 shows assumed parameters and their values used in the experiments. Two design problems were investigated, including the conceptual design of a wind bracing system and the conceptual design of the entire steel structural system in a tall building. 30-story buildings with 5 bays were the subject of design. Seven types of wind bracing members, two types of beams, and two types of supports were considered. The types of column members were kept the same during the entire design process.

WIND BRACING SYSTEMS

In the conceptual design of wind bracing systems, only the configuration of a wind bracing subsystem was evolved. The best designs in terms of the minimum total weight of the steel structure were sought. Figure 7 shows typical results obtained in this group of experiments. It clearly shows that the generative representations based on 1D CAs (both standard and totalistic CA rules) produced significantly better optimization results than traditional parameterized representations. The former representation also found the optimal solutions very quickly (in less than 150 fitness evaluations as shown in Figure 7).

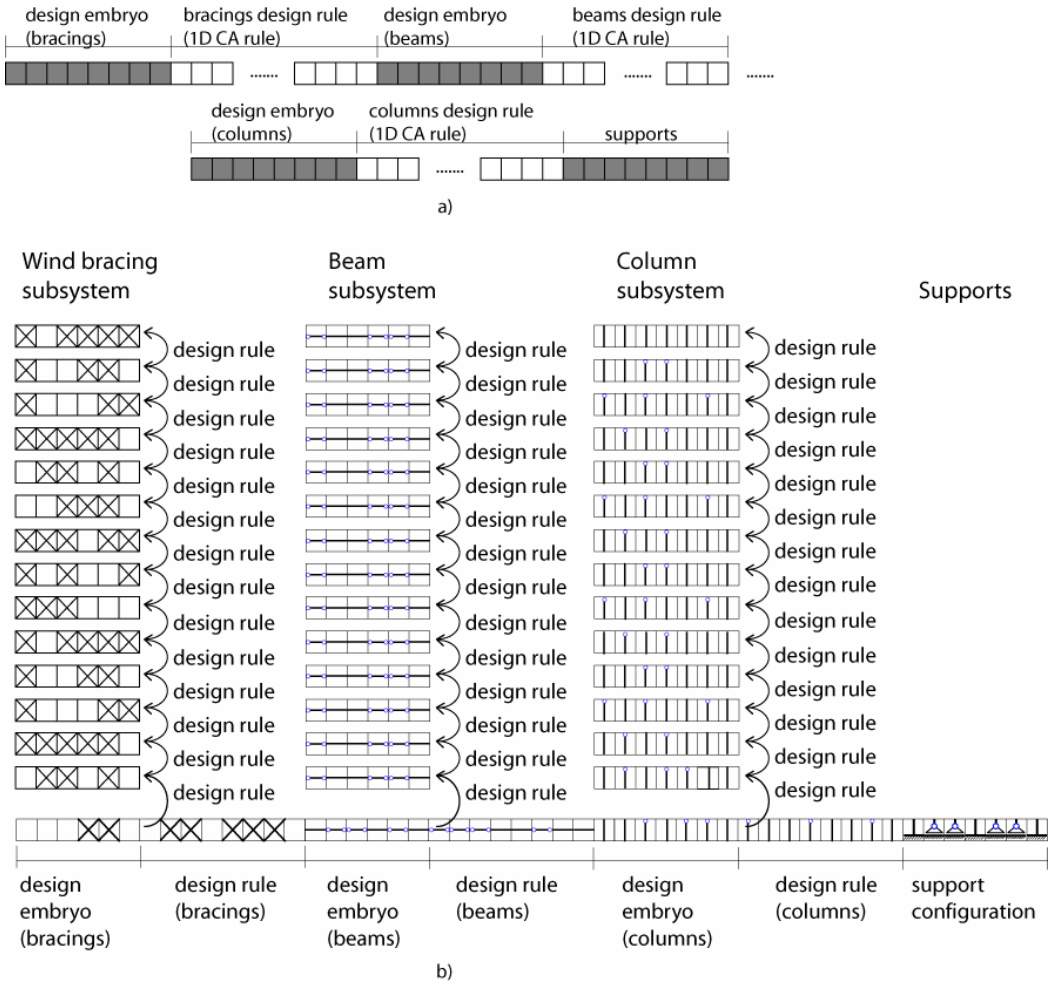

Figure 6: Generative representation of an entire steel structural system in a tall building based on multiple 1D CAs

Table 1: Parameters and their values used in the reported experiments

Domain Parameters	Value(s)	EA Parameters	Value(s)
Number of bays	*5*	Representations	*1D CA, 2D CA, Parameterized*
Number of stories	*30*	CA rule types	*Standard, Totalistic*
Bay width	*6.01m (20 feet)*	Type of EA	*Evolution Strategies*
Story height	*4.27m(14 feet)*	Population sizes	*(1,5), (1,25), (5,25), (5,125)*
Structural analysis	*1st order*	Mutation rates	*0.025, 0.1, 0.3, 0.5*
Types of beams	*Pinned, Fixed*	Crossover rates	*0.0, 0.2, 0.5*
Types of columns	*Fixed*	Fitness	*The total weight of the steel structure*
Types of supports	*Pinned, Fixed*	Number of runs	*5*
Types of wind bracings	*No, Diagonal (/), Diagonal (\), K, V, simple X, X*	Termination criterion	*1,000 fitness valueation (short-term), 10,000 fitness evaluations (long-term)*

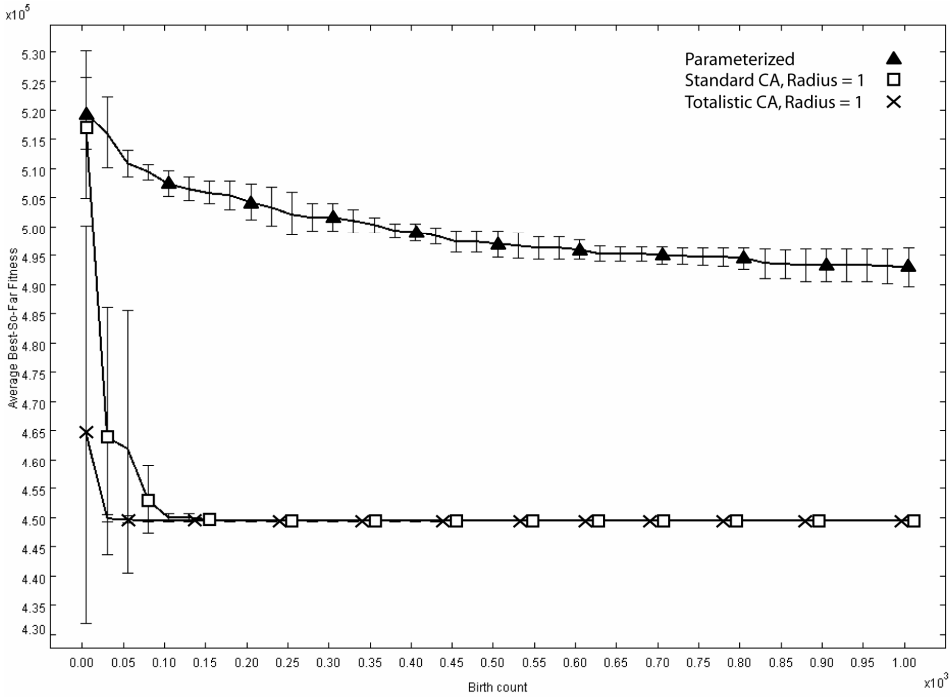

Figure 7: Comparison of evolutionary optimization of wind bracing systems achieved by generative representations (standard CA and totalistic CA) and parameterized representations

Both types of design representations exhibited large qualitative differences in terms of structural shaping patterns of the generated design concepts. The left hand side of Figure 8 presents the structural shaping patterns produced by the generative representations based on 1D CAs. It shows several interesting emergent patterns which are qualitatively different than the fairly random looking configurations generated in the experiments in the parameterized representations (see the right hand side of Figure 8).

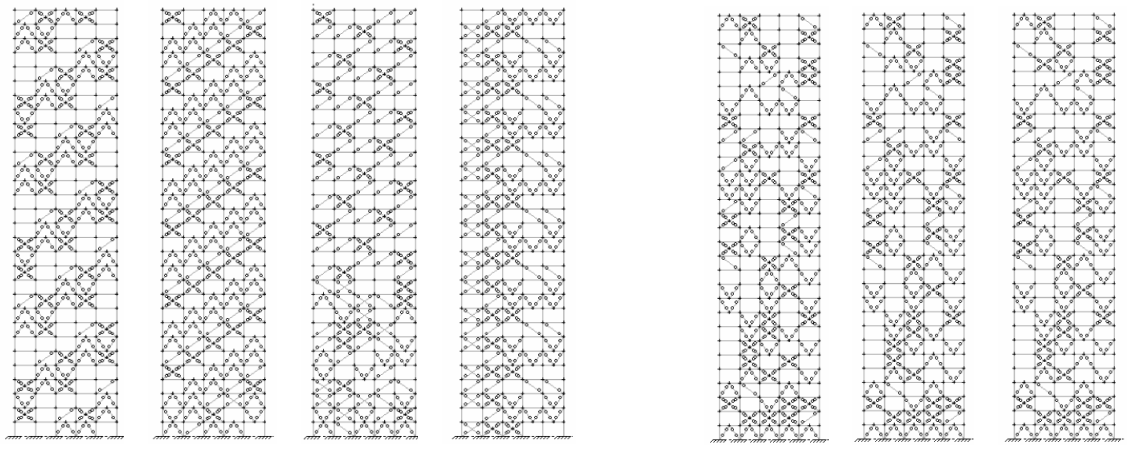

Figure 8: Comparison of structural shaping patterns produced by the generative representations (left) and parameterized representations (right)

ENTIRE STEEL STRUCTURAL SYSTEMS

Figure 9 shows typical results produced in the long-term experiments (10,000 evaluations). As before (see Figure 7), the generative representations significantly outperformed the parameterized representations. Specifically, here the best results were produced by totalistic CA rules which produced the design concepts of minimum weight and found these solutions very fast (in less than 2,000 fitness evaluations). The authors also observed similar qualitative differences in structural shaping patterns produced by generative and parameterized representations (not shown here).

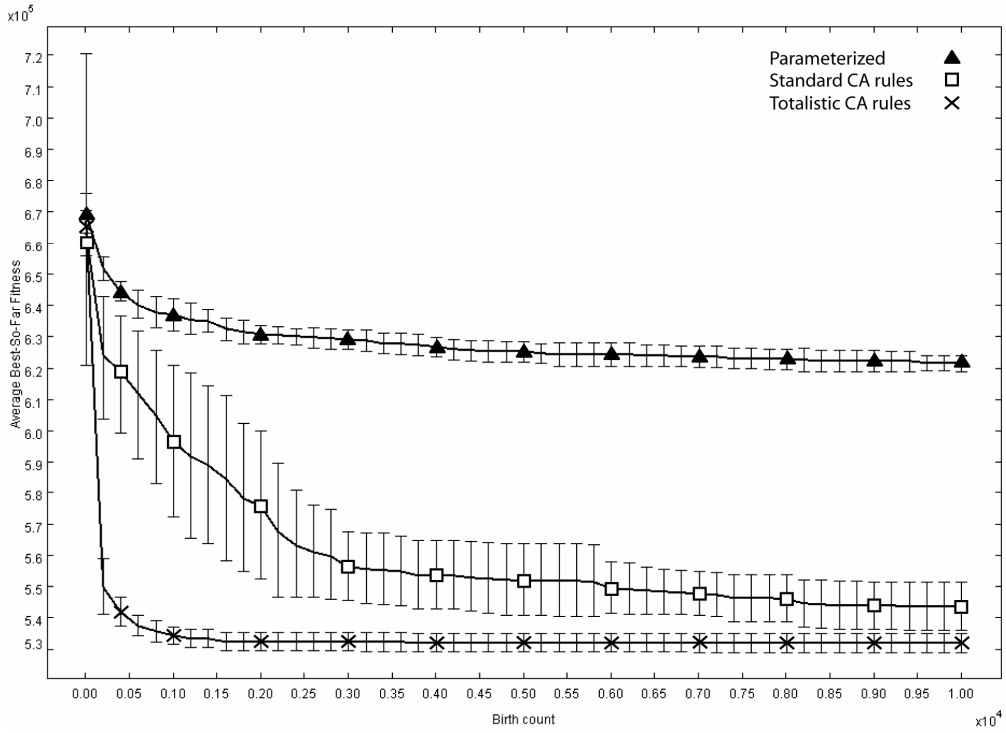

Figure 9: Comparison of evolutionary optimization of entire steel structural systems achieved by generative and parameterized representations

CONCLUSIONS

In this paper two types of generative representations of structural systems were presented and compared to traditionally used parameterized representations. Three examples of generative representations of steel structural systems in tall buildings based on one-, and two-dimensional cellular automata were defined and empirically investigated. The results of design experiments have clearly shown important advantages of the proposed representations. They outperformed the parameterized representations both quantitatively and qualitatively. Specifically, they produced design concepts of significantly improved fitness and exhibiting interesting structural shaping patterns.

The generality of the proposed representations will make them suitable for application in other problem domains. The authors are planning to apply them to other structural engineering problems. They will also investigate more advanced types of CA encodings, e.g. self-adaptive CAs or non-uniform CAs, and apply them to several structural engineering problems.

REFERENCES

Arciszewski, T., and De Jong, K. A. (2001). "Evolutionary computation in civil engineering: research frontiers." Proceedings of the Eight International Conference on Civil and Structural Engineering Computing, B. H. V. Topping, ed., Eisenstadt, Vienna, Austria.

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

Bentley, P. J., and Kumar, S. (1999). "Three ways to grow designs: a comparison of embryogenies for an evolutionary design problem." Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99), W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. J. Jakiela, and R. E. Smith, eds., Orlando, Florida, USA, 35-43.

Hornby, G. S. (2003). "Generative representations for evolutionary design automation," Ph.D. Dissertation, Department of Computer Science, Brandeis University, Waltham, MA.

Inou, N., Shimotai, N., and Uesugi, T. (1994). "Cellular automaton generating topological structures." Proceedings of the Second European Conference on Smart Structures and Materials, A. McDonach, P. T. Gardiner, R. S. McEwen, and B. Culshaw, eds., Glasgow, UK, 47-50.

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.

Kicinger, R., Arciszewski, T., and De Jong, K. A. (2005a). "Emergent Designer: An integrated research and design support tool based on models of complex systems." Artificial Intelligence for Engineering Design, Analysis and Manufacturing(Special Issue on Design Computing and Cognition), submitted.

Kicinger, R., Arciszewski, T., and De Jong, K. A. (2005b). "Evolutionary computation and structural design: a survey of the state of the art." Computers & Structures, approved.

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

Wolfram, S. (2002). A new kind of science, Wolfram Media, Champaign, IL.