Special Invited Lecture CC2005

Structural Design Inspired by Nature[1]

Tomasz Arciszewski
Rafal Kicinger

Department of Civil, Environmental and Infrastructure Engineering
School of Information Technology & Engineering
George Mason University
4400 University Drive MS 4A6
Fairfax, Virginia 22032
USA

Abstract

This paper explores the state of the art in structural design inspired by nature and proposes an improved understanding of this emerging paradigm and its major components. First, it introduces and discusses the nature of three categories of inspiration, including visual, conceptual, and computational inspiration. Next, several important inspiration sources are identified and briefly described, including Evolutionary Computation, Coevolutionary Computation, Cellular Automata, and TRIZ. In particular, design generation mechanisms based on cellular automata are introduced with some details. They are inspired by the processes of morphogenesis occurring in nature and have great potential to generate novel designs. In this case, the design generation mechanisms are encoded in so-called generative representations, which are also described.

Three major design objectives are introduced and discussed, namely optimality, creativity, and robustness. They are related to the sources of inspiration and the corresponding computational mechanisms inspired by nature. Also, three levels of integration of computational mechanisms inspired by nature are proposed and their relationship to design objectives is discussed. Finally, a general design situation, when inspiration by nature is considered, is introduced and its unified description is proposed as the first step in the direction of building a unified approach to structural design inspired by nature. The paper provides initial conclusions and discusses the most promising directions for future research.

Keywords: structural design, evolutionary computation, evolutionary design, coevolutionary design, morphogenesis, generative representations, cellular automata.

[1]*Citation*:
Arciszewski, T. and Kicinger, R. "Structural design inspired by nature." Innovation in Civil and Structural Engineering Computing, B. H. V. Topping, ed., Saxe-Coburg Publications, Stirling, Scotland, 25-48.

1 Introduction

Structural design is in a fascinating period of significant changes, both of quantitative and qualitative nature. These changes are mostly driven by progress in Computer Science and in Design and Inventive Engineering. Growing computational resources available to engineers yield quantitative improvements to the analysis and optimization of complex structural systems. Better understanding of the design process, especially of its conceptual stage, and the development of various computational models for the generation of design concepts, create a potential for the fundamental qualitative changes in design, in which design novelty will be controlled and adjusted in accordance to needs.

The history of our civilization clearly demonstrates that progress in engineering is always driven by demand for new systems. This demand is shaped by the rational needs and by our ambitions and dreams. The combination of rational and emotional reasons is very potent. It forces engineers to look not only for incremental improvements of their designs but also to search for novel designs responding to the emotional needs of our societies. The monuments of our civilization, bridges, tall buildings, exhibition halls, etc., always represent the state of the art in structural engineering, but they also reflect the emotional dimension of our profession. Engineering progress is often driven, or at least strongly influenced, by the emotional factors related to the competition among nations (arms race) or among communities, like building cathedrals in the middle ages or tall buildings or bridges these days.

Our monuments are products of rational thinking, systematic analysis, design, and optimization. However, they would never be created without inspiration. It encourages engineers to change their design paradigms and to create novel engineering systems responding to ever growing engineering requirements and to the emotional needs of our societies. Recognizing the importance of inspiration in design and the relationship between inspiration and design creativity is crucial, particularly in the context of impact of this relationship on the computational approaches to design.

The idea of imitating nature is not new. Since the beginning of our civilization, nature often inspired artists and designers by creating the most fascinating designs known to humankind. Initially, designers simply imitated nature's products in their artifacts or engineering systems. However, with progress in science, we gradually began to understand the actual mechanisms governing nature's processes behind the development and evolution of natural systems and to apply these mechanisms to engineering system design. Nowadays, with the emergence of Information Technology, we can computationally simulate many of nature's essential processes.

The paper's objective is to advance structural design inspired by nature by improving our understanding of this emerging design paradigm and its major components. The paper introduces and discusses three categories of inspiration, including visual, conceptual, and computational inspiration. Inspiration may come from various sources, and several key sources are identified and briefly described including Evolutionary Computation, Coevolutionary Computation, Cellular Automata, and TRIZ. Also, three design objectives are introduced and discussed, namely optimality, creativity, and robustness. They are related to the sources of inspiration and the corresponding computational mechanisms inspired by nature. Also, three levels of integration of computational mechanisms inspired by nature are proposed and their relationship to design objectives is discussed. Finally, a general design situation, when inspiration by nature is considered, is introduced and its unified description is proposed as the first step in the direction of building a unified approach to structural design inspired by nature. The paper provides initial conclusions and discusses the most promising directions for future research.

2 Inspiration and Design

The role of inspiration in design is still poorly understood, especially its relationship to design paradigm changes leading to novel design concepts. In this context, a design paradigm can be understood as a set of design principles/assumptions valid for a large class of designs. Such assumptions may be related to the materials to be used, to the design and construction processes, to the assumed behavioral models, etc. For example, we could distinguish design paradigms called "concrete structures," "reinforced concrete structures," and "pre-stressed concrete structures." In the area of tall buildings, we could distinguish, for example, design paradigms called "rigid steel frames," "braced steel frames," "frame tubes," etc.

Each paradigm change involves replacing the operational body of knowledge, or at least changing the context of the knowledge being used. Therefore, such a change is difficult and requires a significant effort combined with the right inspiration. This process can be compared to leaving a box to search for the out-of-the-box designs. Unfortunately, not all engineers are capable of creating novel designs. It is partially caused by the limitations of our engineering education, mostly focused on the quantitative and numerical aspects of design, but it is also caused by our inability to seek and use proper inspiration.

Inspiration can be described as knowledge outside the problem domain which is necessary and sufficient to conduct a paradigm change and consequently to produce novel designs. For example, to develop novel design concepts for a large-span steel beam, a body of knowledge related to pre-stressed concrete beams could be used as inspiration.

The practice of using inspiration in problem solving is known. In fact, the heuristic method of problem solving, called "Synectics," is based on changing the context of the problem-specific knowledge and on the use of knowledge from outside the problem domain when the problem solving process is stalled and needs a boost. Synectics provides even a specific procedure, called "excursion," to temporarily move away from the problem to a random piece of knowledge in order to use this unrelated knowledge as inspiration in the problem solving. Synectics has been developed mostly for applications outside engineering and its engineering use is difficult. It requires training of "synectors," is time consuming, and can be used only for traditional conceptual design being conducted by humans. For all these reasons, TRIZ, discussed in the section 3.4, is so attractive.

At least three kinds of inspiration can be distinguished in design, and each kind plays a different role. All three provide a spectrum whose understanding is critical for the progress in design science, particularly when concerned about the computational approaches to conceptual design.

"Visual inspiration" is relatively well understood and widely used. In this case, pictures (visuals) of various living organisms, or their systems, are used to create similarly looking engineering systems. For example, a picture of a sea turtle shell can be used to shape a reinforced concrete shell for a large span roof structure in an exhibition building. Visual inspiration can produce useful results. Unfortunately, it requires the involvement of a human designer who knows structural engineering and the theory of elastic shells, and who is able, most importantly, to avoid using inspiration in a wrong context. In such a case, a visual can be incorrectly used to produce a dangerous design. For example, the use of the same sea turtle shell shape to design a shear wall in a tall building may result in a structure excessively sensitive to large vertical forces and may be ultimately dangerously unstable.

"*Conceptual inspiration*" occurs when a structural engineer uses a principle found in nature in design, for example, the biological principle of homeostasis. This principle states that any living organism reacts correspondingly to recover its vital functions when attacked by an external agent [1]. A designer can apply this principle, for example, to determine the optimal shape of shell roofs subjected to thermal and mechanical loads (see various shapes of shell roofs reported in [1]). Unfortunately, using conceptual inspiration requires a solid understanding of both nature and structural engineering and cannot be used in a mechanistic way by an automated designing system.

Visual inspiration is skin-deep. Conceptual inspiration is abstract and difficult to use. In fact, both require the involvement of a sophisticated human designer. Fortunately, the third kind of inspiration, called here "computational inspiration," is the most promising from the perspective of automated conceptual design. It is the most intriguing, still poorly understood and difficult, but has the greatest potential to revolutionize design. In this case, inspiration occurs on the level of computational mechanisms, which are inspired by the mechanisms occurring in nature. Such mechanisms are discussed in the following sections.

3 Sources of Inspiration

3.1 Evolutionary computation

Evolutionary computation (EC) is a modern search method inspired by the processes of evolution and natural selection. It encompasses computational models of mechanisms of reproduction, variation, natural selection, and genetic inheritance to solve problems in many fields of engineering and science.

EC, as all methods discussed in this paper, has a strong resemblance to biological processes occurring in nature and is a good example of computational inspiration. It borrows from genetics, evolutionary theory and cellular biology. Thus, a candidate solution to a problem is called an *individual* while an entire set (superset) of current solutions is called a *population*. The actual representation (encoding) of an individual is called its *genome* or *chromosome*. Each genome consists of a sequence of *genes*, which correspond to attributes that describe a solution. When individual solutions are modified to produce new candidate solutions they are said to be *breeding* and the new candidate solution is called an *offspring* or a *child*. Each generated solution is evaluated and receives a grade called *fitness*, which indicates the quality of the solution in the context of a given problem. When the current population is replaced by offspring, the new population is called a new *generation*. Finally, the entire process of searching for an optimal solution is called *evolution* [2].

During more than forty years of EC research, a large variety of evolutionary algorithms (EAs) has been developed. They all share a common set of underlying assumptions but differ in the breeding strategy and/or representation on which they operate [3]. Three major EAs include evolution strategies (ES) [4,5], evolutionary programming (EP) [6], and genetic algorithms (GAs) [7]. There are also many hybrid models incorporating various features of the above paradigms. An excellent introduction to the field can be found in [8].

From the engineering perspective, computationally simulated evolution can be considered as a search and optimization process in which a population of solutions (e.g. designs) undergoes a process of gradual changes. This process is guided by the fitness of individual solutions which is typically defined by a single or multiple objective function(s) (e.g. the total weight of a structural system, etc.).

The structure of a canonical EA is presented in Figure 1 [9]:

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

Figure 1: A canonical evolutionary algorithm

Before an actual evolutionary process begins, an initial population of individuals (designs) is created. This is usually done randomly but several other initialization methods may also be utilized (e.g. starting from a set of previously known designs). Each individual in the initial population is evaluated and assigned a fitness value. Based on these fitness scores, the selection mechanism chooses a subset of the current population as parents to create new individuals. These new individuals are created by copying the parents and applying variation operators. The two most popular variation operators include mutation and recombination. Mutation acts on a single individual and works by applying some variation to one or more genes in the individual's chromosome. Recombination, on the other hand, operates on multiple individuals (usually two) and combines their parts.

The newly created individuals (offspring) are evaluated and assigned fitness values. Then, either the entire population of parents or its subset is replaced by these new individuals. Steps 3-6 in Figure 1 are repeated until an assumed stopping criterion is met, which is usually defined as an arbitrary number of generations or fitness function evaluations.

3.2 Coevolutionary computation

Coevolutionary computation is an important branch of EC research that has recently received significant attention. As before, the inspiration for coevolutionary algorithms comes from biological processes encountered in many natural ecosystems. Coevolutionary models typically include two or more populations which simultaneously evolve and where no *objective* fitness function exists. On the contrary, each individual's fitness is a *subjective* function of its *interactions* with individuals from coevolving populations [10,11].

Initial computational models of coevolutionary behavior were formulated by Maynard Smith [12] and Axelrod [13,14]. They were further extended by Hillis [15] and others and resulted in a new class of algorithms called coevolutionary algorithms (CEAs). Two general classes of coevolutionary models have been proposed: competitive and cooperative. Competitive coevolutionary models are especially suitable for problem domains where it is difficult to explicitly formulate an objective fitness function, for example in game-playing strategies, etc. Such models have been applied in engineering, e.g., to solve constrained optimization problems [16]. On the other hand, cooperative coevolutionary models [17] have been used for problem domains where explicit notions of modularity could be introduced [18].

Wiegand [10] identifies 7 major attributes defining possible coevolutionary architectures (see Table 1). They describe ways in which coevolutionary systems may be set up. The attributes include the payoff quality, methods of fitness assignment, methods of interaction, update timing, problem decomposition, spatial topology, and population structure. A detailed description of these attributes and their values can be found in [10].

Coevolutionary models provide great possibilities for many engineering design applications. For example, traditionally complex engineering design problems can be decomposed (modularized) into simpler problems which are solved independently. This approach proved to provide good results for engineering problems which could be linearly decomposed (the superposition principle). This is no longer the case, however, for complex designs with nonlinear interactions and dependencies among subcomponents. In these situations, cooperative coevolutionary models provide potentially more appropriate framework for designing such types of systems because they allow for an explicit specification of interactions among subcomponents and make no assumptions regarding their linear character. Moreover, in cooperative coevolutionary models subcomponents of complex engineering systems undergo dynamic co-adaptation processes which improve their robustness.

Table 1: Attributes describing coevolutionary architectures [10]

Attribute	**Attribute Values**		
Payoff quality	Cooperative	Competitive	Non-competitive
Methods of fitness assignment	Implicit	Explicit	
Methods of interaction	Sample size	Selective bias	Credit assignment
Update timing	Sequential	Parallel	
Problem decomposition	Partitioning methods	Temporal decomposition	
Spatial topology	Spatial embedding	Non-spatial embedding	
Population structure	Single	Multiple	

So far, only initial experimental studies have been conducted using coevolutionary models in engineering design, particularly in architectural design. Maher and Poon [19] suggested that it is often the case in a design process that requirements are reconsidered when a design solution is offered. They introduced the idea of coevolutionary design in which requirements and solutions coevolve in two separate populations. The first population contains design solutions while the second includes design requirements. The fitness function evolves with the requirements and is different (local) at various stages of the coevolutionary design process. In structural design, Nair and Keane [20] used cooperative coevolutionary algorithms in structural optimization of cross-sections of members of planar truss systems. The optimized truss systems were decomposed and coevolved in separate populations. Shelton [21] is working on coevolution in the context of conceptual design for robustness of complex engineering systems with examples from the domain of satellite systems. His research has a fundamental nature and its results and design tools being developed should be applicable to structural design.

3.3 Cellular Automata

Cellular automata (CAs) are simple mathematical representations of complex systems. As other computational models presented in this paper, they were proposed as models of processes and phenomena occurring in nature. CAs were introduced in the early 1950s by von Neumann [22] as a possible model for biological systems. Since that time, they have been successfully applied in physics, biology, chemistry, economy, geology, and other disciplines. CAs have been used to model complex systems and processes consisting of a large number of identical, simple, and locally interacting components. Following Ilachinski [23], we can distinguish 5 basic characteristics of CAs:

- *Discrete lattice of cells*: the system consists of usually 1-, 2-, or 3-dimensional lattice of cells (higher dimensional extensions are also possible but rarely used in practice).
- *Homogeneity*: all cells are equivalent.
- *Discrete states*: each of the cells can be in one of the finite number of possible discrete states.
- *Local interactions*: each cell interacts only with cells contained in its local neighborhood.
- *Discrete dynamics*: at each discrete time unit, each cell updates its current state according to a transition rule taking into account the states of cells in its neighborhood.

One of the characteristic features of CAs is their ability to exhibit complex patterns of behavior using a set of simple underlying rules. They appear to capture many essential features of complex, self-organizing, and *emergent* behavior observed in many natural systems.

CAs can be uniquely defined by their transformation rules (also called CA rules) and their initial configurations of cells. A CA rule is understood here as a complete set of decision rules whose conditions incorporate all possible combinations of cell values in the given local neighborhoods. The outcomes of these rules determine the values of the considered cells (usually central cells in each 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). The value of each cell at a next time step is determined by a CA 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.

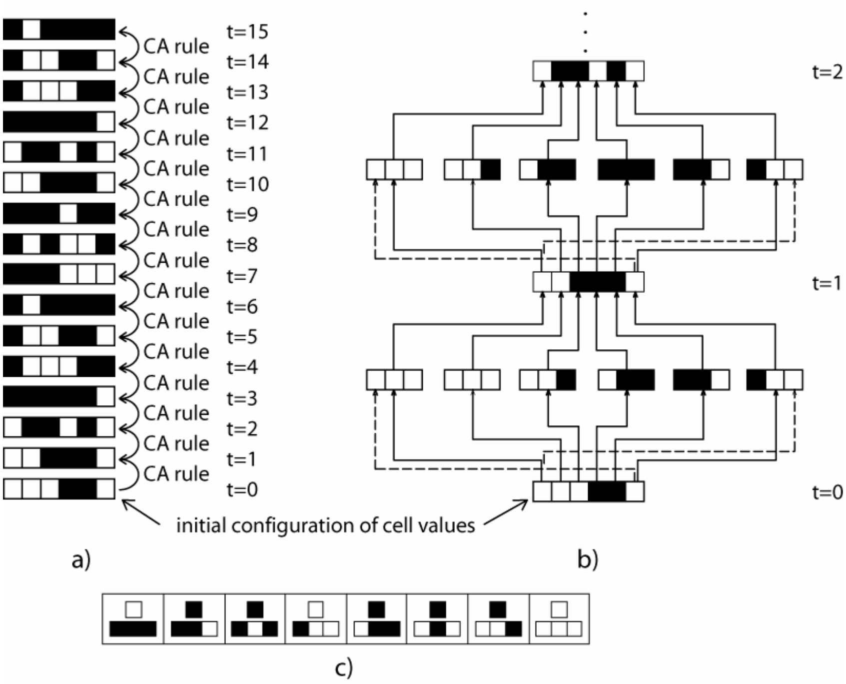

Figure 2: Process of iteration of an elementary cellular automaton

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 periodic 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 CA rule). 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.

CAs offer enormous and still largely unexplored opportunities for structural design. First, they are well-suited to model spatial relations among various elements of a structural system. For example, in the context of a steel structural system, CA rules can model spatial interdependencies among various bracing elements. Second, the shape and size of CAs' local neighborhoods can be explicitly used to represent the directions and range of local interactions among elements of a structural system. Finally, CAs are known to produce various kinds of emergent behavior. This property may be very important for many structural design applications, e.g. when novel emergent designs/patterns are sought.

Existing applications of CAs in structural design investigated self-organization of topologies in mechanical structures [24], shape optimization of structural plates [25], and shape and topology optimization of two-dimensional elastic structures [26]. Hajela and Kim [27] applied EAs to search the space of CA rules in structural analysis of 2D elastic structures.

3.4 TRIZ

During the last 50 or so years, Altshuller [28] and his followers [29-31] have developed a family of problem solving methods in the context of engineering design, called "TRIZ." All these methods are based on two fundamental assumptions. First, a novel or creative design concept is obtained when a technical contradiction is eliminated. In this case, a technical contradiction is understood as an interrelated pair of abstract characteristics of an engineering system, when one characteristic is improved, the second one is worsened, for example, rigidity of a structural member versus its weight. The second assumption is that a technical contradiction can/should be eliminated using a proper piece of engineering knowledge (inspiration). Depending on the nature of knowledge used in the process, several categories of resulting design concepts can be distinguished, as shown in Table 2.

Depending on the category of design concepts to be produced, various conceptual and computational operations are necessary and they are listed in the column 4, Table 2. The individual operations are briefly discussed here in accordance to [32] and [33]. The purpose is to relate the operations to various forms of computational inspiration, as proposed in previous sections.

1. Selection

"A design concept is selected from a group/class of known concepts in a given engineering domain."

This operation can be interpreted as a case when an EA is using only the selection mechanism and the process of evolution is initialized with a population of known design solutions, rather than with randomly generated ones.

2. Modification

"A design concept is produced as a combination and/or modification of known design concepts from a given domain. The modification process can be performed either deterministically or using a random generation process."

In this case, an EA is searching for an optimal solution in a parameterized representation space of a class of engineering designs in a specific domain, for example in the domain of steel beams.

Table 2: Knowledge-Based Categories of Design Concepts

Design Concept Category	**Concept Description**	**Inspiration**	**Knowledge Operation**
1	**2**	**3**	**4**
Standard or Apparent Concept	Well-known and simply selected from a class of known concepts in a given domain	None	Selection
Improved Concept	A combination of known concepts in a given domain	None	Modification
Innovative Concept or Invention within Paradigm	A combination of known concepts from two different but related domains	Yes Example: in steel design the use of inspiration from concrete design	Innovation
Inventive Concept or Invention outside Paradigm	A concept developed using knowledge from at least two much different domains	Yes Example: in steel design the use of inspiration from biology	Invention
Discovery	A concept based on a new scientific principle	Yes Example: a concept of an x-ray machine based on the discovery of x-rays	Discovery

3. Innovation

"A design concept is produced as a combination of known concepts from a given domain and other domains."

This operation can be interpreted as the island-model EA where various populations of designs evolve independently and occasionally exchange some individuals through a migration process. The migrations can model the introduction of knowledge from other domains to a particular engineering domain. For example, one population may represent a class of steel beams while the second one represents a population of reinforced concrete beams.

4. Invention

"A design concept is produced as a combination of known concepts from a given domain and new concepts based on a new technology, which have been recently introduced."

A good example of such design concept is the concept of a ceramic disc brake, recently developed by Porsche. In this case, the well-known concept of a disc brake from the area of mechanical engineering is combined with a concept of a ceramic disc. This concept is based on the recently developed technology of high strength and temperature resistant ceramic compounds, a product of the materials science.

In the case of invention, EC is used to evolve both the values of attributes and the attributes themselves. A combination of transformation operators [32] is used for a representation space. It may include attribute addition (introduce new attributes/genes to the representation space), attribute elimination (removing unimportant attributes), attribute abstraction (combining attributes into larger units, or components, and subsequently exploring the component based representation [34]), and attribute construction (creating new attributes by a simple or complex transformation of the initial attributes).

5. Discovery

"A design concept is produced as a combination of known concepts from a given domain and new concepts based on new scientific principles."

A classical example of such a concept is the concept of an x-ray machine. It is based on a number of well-known mechanical and electrical engineering concepts combined with a new concept of seeing through the human body utilizing the new scientific principle of x-rays.

Unfortunately, at this time, we simply do not know any computational operators capable of producing discoveries. However, special types of representations, namely the generative representations [35], seem to be necessary to accomplish it. Generative representations use compact representations (genotypes) of existing design knowledge and mappings that translate these representations to actual designs (phenotypes). The mappings can reuse elements of the representations during the process of translation. Thus, the compact representations can be thought of as storing existing knowledge on a given engineering domain, whereas mappings correspond to new scientific principles that can transform the known concepts to new, and possibly creative, design concepts. The mappings are usually simple programs that take the compact representations as input and produce actual design concepts as output. Despite their simplicity, they can generate designs that can be defined as creative [36]. Recently, Wolfram [37] suggested that all scientific principles and natural processes can be modeled in terms of simple programs that can nevertheless produce complex behavior. EA using the generative representations will search both the space of compact representations and the space of simple transformation programs (scientific principles) and will generate creative design concepts.

In engineering the focus is on producing innovative and inventive concepts, which obviously require inspiration. Unfortunately, the table also demonstrates that it is virtually impossible for a single engineer to acquire knowledge from several domains, which would be sufficient to routinely produce innovative and inventive concepts. Therefore, an abstract engineering knowledge is necessary, particularly related to the elimination of technical contradictions, and such knowledge is provided by TRIZ.

4 Design for Optimality

The sources of computational inspiration from nature discussed in the previous section can be used to address various objectives of engineering design. In this paper we focus on three important objectives: optimality, creativity, and robustness. In order to be able to use any of these mechanisms and conduct design activity on a computer we first need to properly represent the design problem in a form which is suitable to for computational processing.

In this paper, we consider representations of design problems in the context of three major engineering design objectives:

Objective 1: CREATIVITY

In this case, the objective is to generate unknown yet feasible design concepts.

Objective 2: OPTIMALITY

The focus is on the generation of optimal design concepts with respect to assumed optimality criteria.

Objective 3: ROBUSTNESS

The objective is to produce robust designs, i.e., designs whose performance is relatively insensitive to changes in the environment.

Below, we individually discuss computational models inspired by nature which can be applied at each of these levels and later in section 7 we show how they can be combined and form an integrated design framework inspired by nature.

Evolutionary computation was among one of the first nature-inspired computational models of design processes. Computationally simulated evolution was viewed as a natural model of design process in which an engineering system being designed is gradually improved with respect to a single, or multiple, objective(s) (Objective 2: Optimality) [38]. Initial applications of EC in structural design were mainly focused on sizing and shape optimization in the context of specific design concepts. Later, more complex conceptual design problems were investigated where typically optimal topologies of structural systems were sought [33].

In these strictly optimization-oriented applications, the computational models inspired by nature were limited to representing design processes. In the case of the other two objectives, however, traditional models were employed. For example, Figure 3 shows a typical parameterized representation of a structural system in a tall building (Objective 1).

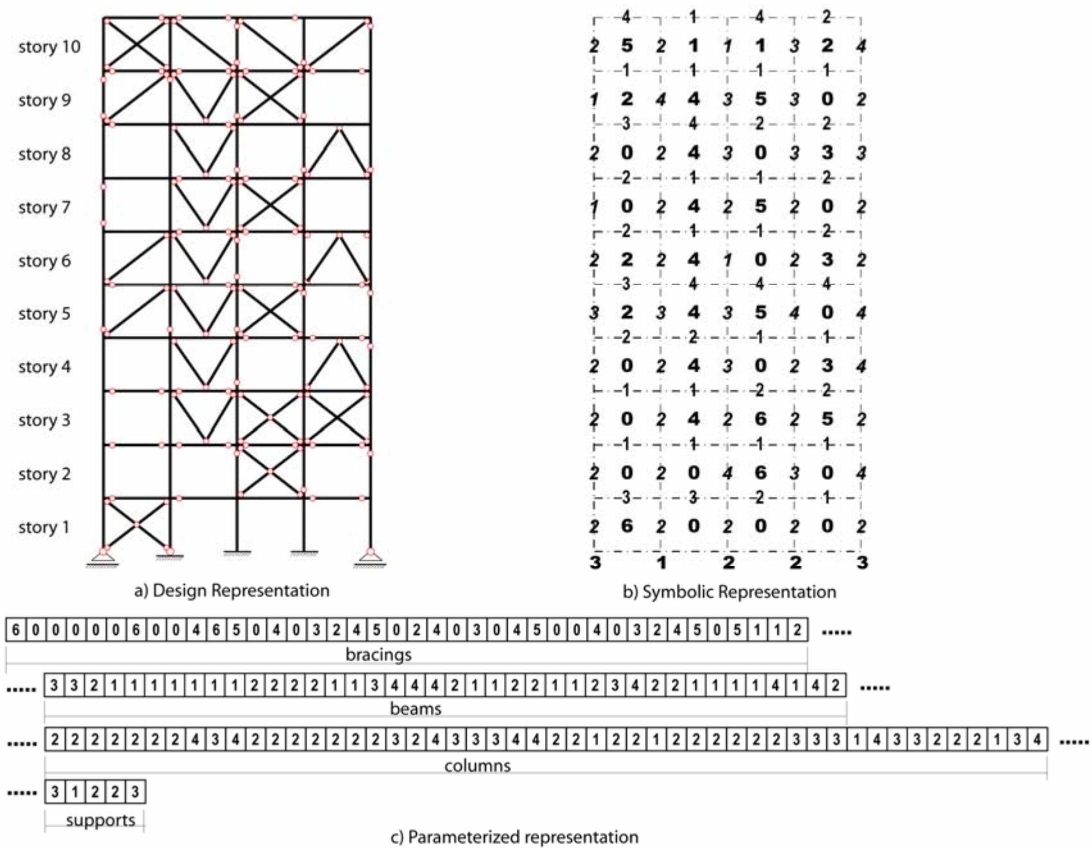

Figure 3: Typical parameterized representation of a structural system in a tall building used in evolutionary optimization applications [39].

In this case, the system (see Figure 3a) is represented by a set of symbolic attributes (see Figure 3b) which are encoded in a linear genome (see Figure 3c). Similarly, for Objective 3 typical loads and load combinations were applied. Their character, locations, and magnitudes were determined by appropriate design codes.

Evolutionary algorithms proved to perform very well for a large spectrum of design optimization problems, especially those involving nonlinear, stochastic, temporal, or chaotic components. They were, however, inadequate when another important objective of engineering design was sought, namely creativity. Nature inspired mechanisms that can remedy this problem are presented in the next section.

5 Design for Creativity

Traditional methods for representing engineering designs proved to work well for design optimization tasks. They didn't, however, produce satisfactory results when novel designs of structural systems were sought [33] (Objective 1: Creativity). Consequently, several researchers proposed alternative ways of representing designs inspired by biological processes of morphogenesis [35,36,40]. These investigations led to a concept of morphogenic design [40,41] illustrated in Figure 4. Here, instead of a traditional direct mapping of elements of the structural system into attributes/genes (as in Figure 3), a design is developed, or 'grown,' from an initial seed (called here a 'design embryo') similarly as it is done in nature, e.g. in plants. These representations, often called generative representations, do not encode entire design concepts, as in parameterized representations, but rather rules on how to build these designs from initial design embryos.

Cellular automata, introduced in section 3.3, can be used as computational models to simulate the processes of morphogenesis in nature. The initial configuration of cells in a CA constitutes an embryo. The structure is developed, or grown, from the embryo by repeatedly applying the CA rule. A graphical representation of a morphogenic design process is presented in Figure 5. Here, a design concept of a steel structural system in a tall building is grown from its design embryo (initial configuration of cells in Figure 5a) by applying the design rule based on a CA rule (see Figure 5b). Each application of the design rule defines a configuration of a subsequent story in a tall building. This process is repeated until the structure of the entire structural system is fully developed (see Figure 5c).

Several types of generative representations based on cellular automata have been studied in [41]. They were shown to produce qualitatively different design concepts which exhibited interesting and emergent shaping patterns.

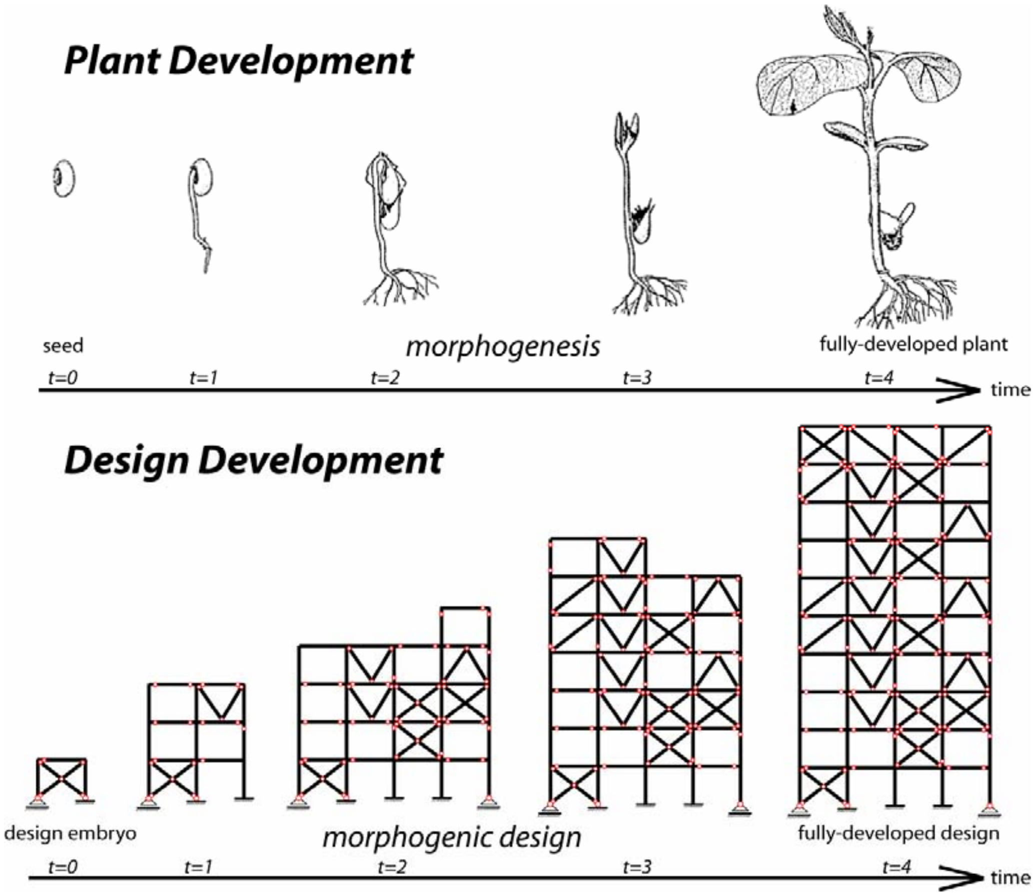

Figure 4: Process of development of a steel structural system in a tall building inspired by the processes of morphogenesis occurring in nature

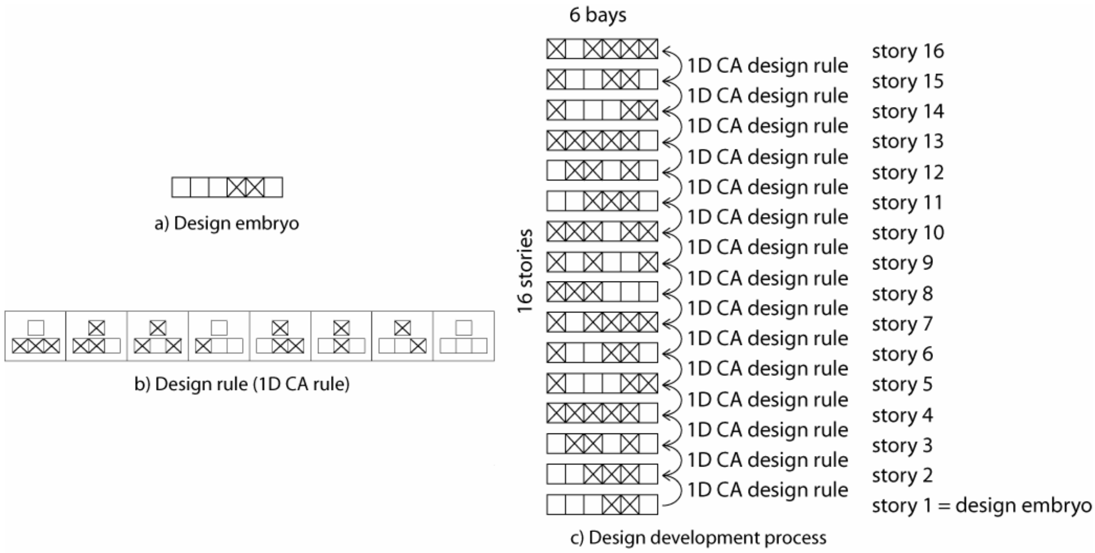

Figure 5: Generative representation of a steel structural system in a tall building based on one-dimensional cellular automata

6 Design for Robustness

One of the characteristic properties of biological systems is their adaptive behavior, i.e. their underlying mechanism to adapt and survive in uncertain environments (Objective 3: Robustness). Hence, they are often referred to as complex adaptive systems that can appropriately adapt to the environmental changes. From an engineering point of view it is important to ensure that engineering designs can adapt to changing environmental conditions because that guarantees their robustness, a required property of almost all engineering products. In the context of structural design, for example, the environmental conditions can be understood as locations, types, and magnitudes of loads applied to the system during the design evaluation process.

Traditionally, only a limited number of loading cases and load combinations were considered in the design evaluation process, usually determined by appropriate design codes. This proved to work well for structural systems operating in typical conditions. Nowadays, however, designers often need to consider dramatically different types of loads which may be applied to structural systems, for example blasts, explosions, intentional fires, etc. In these situations, it becomes problematic to determine most harmful locations and magnitudes of these types of loads (also combined with traditional loads) and estimate robustness of the structural systems subjected to them.

Nature inspired coevolutionary models, described in section 3.2, can be used to overcome deficiencies of traditional methods. We can build computational representations of design evaluations processes in which two competing populations are considered:

- a population of structural designs, and
- a population of loads

They coevolve in the following way. The fitness of each individual design in the population of designs is determined by measuring how well it performs against the loading cases from the population of loads. On the other hand, fitness of each loading case will depend on the number of designs it "defeated," i.e. how many designs didn't succeed to satisfy design requirements (like max. bending moment, max. displacement, etc.) under this loading case.

This approach may be helpful in testing sensitivity of various structures to certain classes of loadings that would never be applied by human designers. There is, however, a lot to be done to understand the true potential of this paradigm in structural design. So far, very little has been done in this area and it is potentially one of the most promising paths of future research.

7 Nature Inspired Design – A Unified Description

So far, we have discussed how various computational models inspired by nature can be used in structural design to address its important objectives. In this section, we demonstrate ways in which they can be integrated, both semantically and computationally, and how a design situation should be understood to properly consider inspiration provided by nature.

Addressing both the creativity and optimality objectives in engineering design is a desirable goal. In order to achieve it, mechanisms of design concept generation and optimization need to be properly integrated. It turns out that it is feasible when computational mechanisms inspired by nature are used.

Figure 6 shows a simple example of how generative representations based on cellular automata can be encoded as genomes and manipulated by evolutionary algorithms. Figure 6a presents a schematic structure of such genome consisting of two major parts: encoding of the design embryo and encoding the design rule. Figure 6c shows how the design embryo and the design rule can be encoded in a genome manipulated by an EA. The actual numeric form of this genome is presented in Figure 6b. A detailed description of this and other encoding schemes can be found in [41].

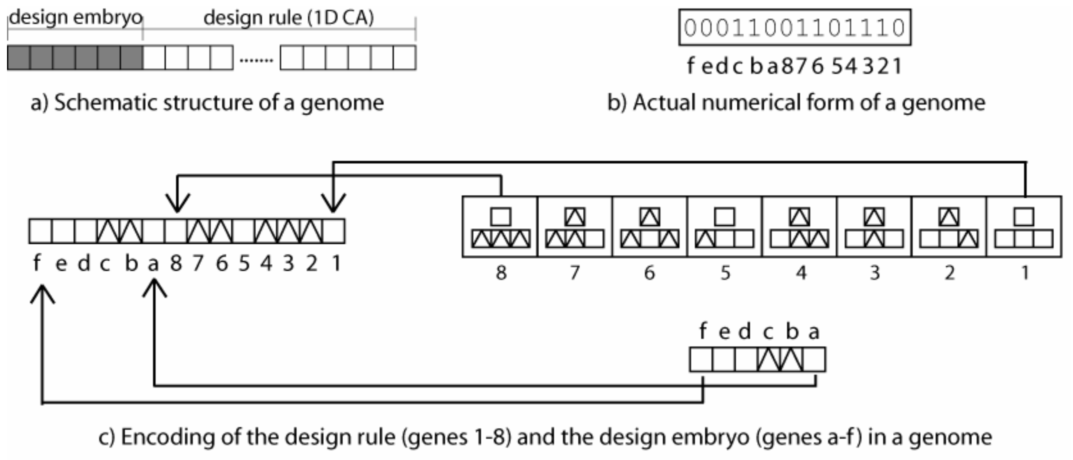

Figure 6: Integration of cellular automata representations and evolutionary algorithms in morphogenic evolutionary design [40]

Extensive experimental studies conducted with cellular automata combined with evolutionary algorithms (named *morphogenic evolutionary design* [40,41]) have shown that EAs more efficiently search the space of design rules and design embryos compared to traditional parameterized representations. At the same time, they generate interesting structural shaping patterns which are qualitatively different than patterns produced using parameterized representations [40,42,43].

In a similar way, cellular automata can be combined with coevolutionary algorithms and hence achieve two other important objectives, creativity and robustness. Unfortunately, at this time very little is known when and how various representations and evolutionary, or coevolutionary, algorithms could, or should, be used together to achieve the desired results. For this reason, three levels of integration of mechanisms inspired by nature are proposed here, as a general framework for understanding this important issue. Figure 7 presents the levels of integration of computational mechanisms inspired by nature and their relationships to design objectives.

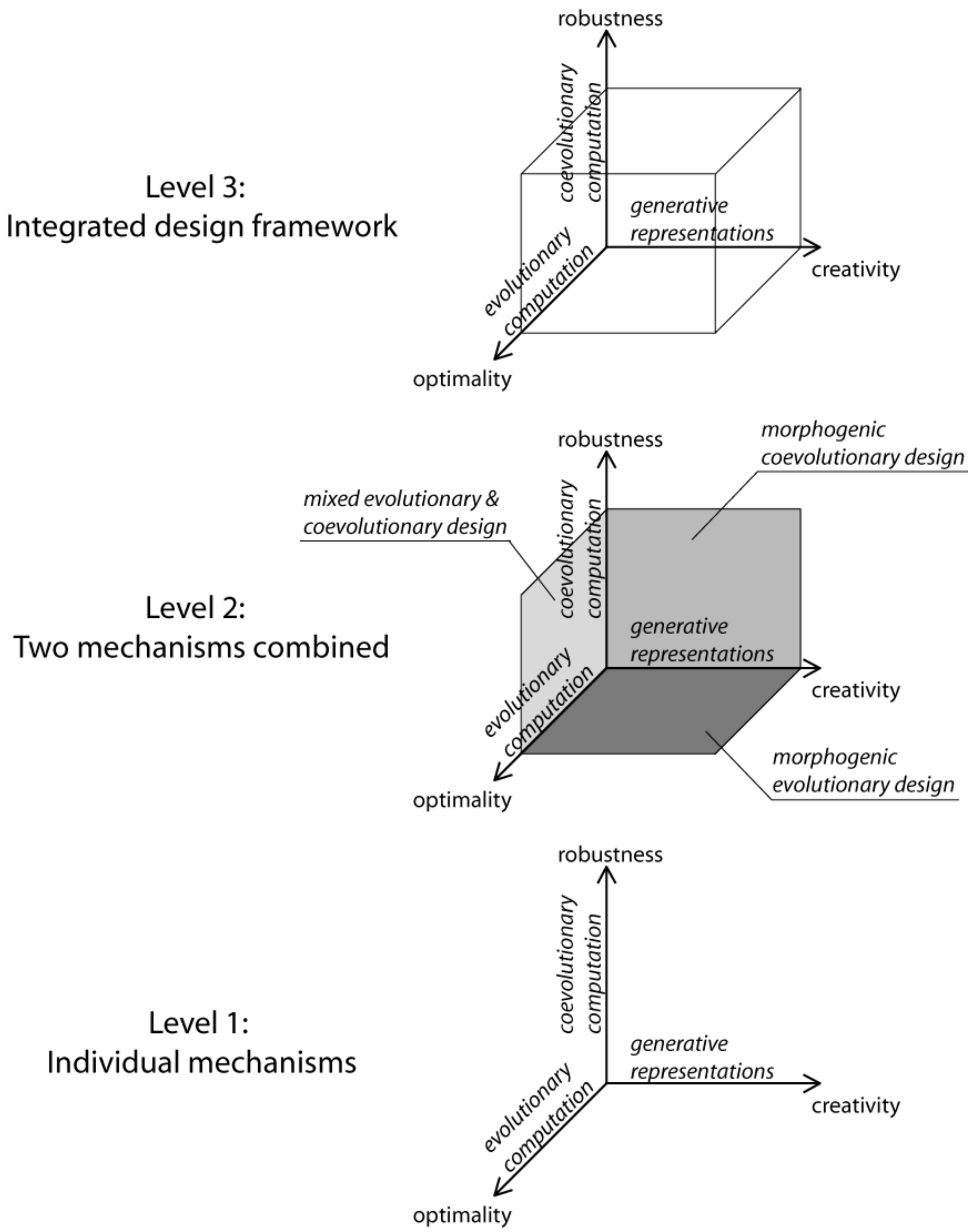

Figure 7: Levels of integration of mechanisms inspired by nature and their relationship to design objectives

At the lowest level of integration (Level 1), individual mechanisms are used separately to achieve corresponding design objectives. For example, evolutionary computation is employed to optimize structural systems and generative representations (e.g., cellular automata) to generate novel designs. Thus, this level can be called 'one-dimensional' design because it only considers a single design objective at a time. A relatively large body of knowledge is available at this level, particularly in the case of evolutionary based optimization [33].

At a higher level (Level 2), two mechanisms inspired by nature are combined to address two design objectives. For example, evolutionary computation and generative representations can be integrated to generate novel designs concepts and subsequently optimize them (see morphogenic evolutionary design discussed above). Similarly, integration of coevolutionary algorithms and generative representations yields morphogenic coevolutionary design addressing robustness and creativity objectives. It is also possible to combine evolutionary and coevolutionary models in a mixed evolutionary and coevolutionary design. In this case, optimal designs are produced first and subsequently tested by various coevolved evaluation scenarios. This level of integration, the 'two-dimensional' design, corresponds to current state-of-the-art in nature inspired design research and hence relatively small body of knowledge is available on the subject.

Finally, at the highest integration level, the three computational mechanisms discussed in this paper are combined in an integrated nature-inspired design framework. This 'three-dimensional' design constitutes a useful reference for future research in engineering design and building future design support tools. Such tools are necessary to provide guidance through the structural design process and acquire relevant design knowledge.

From the functional point of view, a future design support tool should analyze first a given design situation to identify the design objectives and their relative importance. Based on these results, it would recommend the type of inspiration to be used, the most appropriate type(s) of representations, and the computational mechanisms. Unfortunately, such a tool has yet to be developed and it will be a long, research-driven process. This process can be considered as a knowledge acquisition process. In the process must be used a formal description of a given design situation in the form of attributes and their values. Results of an attempt to identify such attributes are provided in Table 3 and called a "unified description of structural design inspired by nature."

The proposed description clearly reveals the complexity of a situation in which six attributes are necessary, each having from three to five values. Unfortunately, progress in structural design requires good understanding how the individual attributes and their values are related. Only if that is accomplished, we will be able to use in a rational and effective way inspiration by nature. The knowledge acquisition process will begin with the generation of data, which in this case will be the result of still to be conducted design experiments performed with various combinations of inspiration, representations, computational mechanisms, etc. The obtained data will be used to acquire knowledge in the form of decision rules describing various relationships among the attributes.

Table 3: Unified description of structural design inspired by nature

Attributes	**Attribute Values**				
Design Objective	Optimality	Creativity	Robustness		
Design Concept	Standard	Improved	Invention within Paradigm	Invention outside Paradigm	Discovery
Inspiration	Visual	Conceptual	Computational		
Representation	Parametric	Cellular Automata	Generative		
Knowledge Operation	Selection	Modification	Innovation	Invention	Discovery
Computational Mechanism	Cellular Automata	Evolutionary Computation	Coevolutionary Computation		

8 Conclusions

The last century of structural design can be characterized by the focus on numerical aspects of the design process. For this reason, the design research objective was mostly to improve analytical methods and tools, which would support the analysis, dimensioning and optimization of structural systems. The recent progress in Computer Science and in Design and Inventive Engineering has begun changing the situation with growing interest in the conceptual design and in the utilization of novel computational mechanisms. In this context, structural design inspired by nature seems to be particularly attractive as combining results of progress in several disciplines. It offers potential for a major breakthrough in our understanding of the structural design process, and that may ultimately lead to a new generation of structural design methods and tools.

The key to the structural design inspired by nature is understanding the concept of inspiration in terms of knowledge and its utilization. Inspiration may come from various sources, and several key sources are identified and briefly described in the paper. The authors believe that appropriate integration of these mechanisms and models is particularly important for design researchers for several reasons. First, there is still very little known about their true potential and limitations. Next, they represent computational inspiration on the most fundamental level. Therefore, the investigation of nature inspired design should bring improved understanding of structural conceptual design on the fundamental level, similar to our understanding of chemical processes design on the level of unit operations, and that would constitute a major breakthrough.

When various sources of inspiration are considered together, they represent an entire spectrum of knowledge, which can be rationally used according to design objectives. Unfortunately, at this stage of research we still do not have full understanding of how various sources of inspiration should be used in design to produce the desirable results. The authors postulate, however, the initiation of research on the subject. Considering the importance of the problem and significant resources necessary to conduct research, an international project might be a good possibility.

The paper introduces and discusses three major design objectives, including optimality, creativity, and robustness. These objectives are related to computational mechanisms inspired by nature, including Evolutionary Computation, Cellular Automata, and Coevolutionary Computation, respectively. The research issue is how to use these relationships in design and how to inspire research, which would improve our understanding of the relationships. The unified description of a design situation is the first step in the direction of a unified structural design model, when inspiration by nature is used. When such a model is available the new structural design paradigm could be better understood, taught, and used in practice.

References

[1] Andres, O. A., Ortega, N. F., & Paloto, J. C. (2005). The homeostatic model as a tool for the design and analysis of shell structures. In M. W. Collins, M. A. Atherton & J. A. Bryant (Eds.), *Nature and design*. Southampton, Boston: WIT Press, 145-156.

[2] Luke, S. (2000). *Issues in scaling genetic programming: breeding strategies, tree generation, and code bloat.* Ph.D. Dissertation, Department of Computer Science, University of Maryland, College Park, Maryland.

[3] De Jong, K. A. (2005). *Evolutionary computation: a unified approach*. Cambridge, MA: MIT Press.

[4] Rechenberg, I. (1965). *Cybernetic solution path of an experimental problem* (Vol. Library Translation 1122). Farnborough, UK: Royal Aircraft Establishment.

[5] Schwefel, H.-P. (1965). *Kybernetische Evolution als Strategie der experimentelen Forschung in der Stromungstechnik.* Master's thesis, Hermann Föttinger Institute for Hydrodynamics, Technical University of Berlin.

[6] Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial Intelligence through simulated evolution*. Chichester, UK: John Wiley.

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

[8] Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. Berlin, New York: Springer-Verlag.

[9] Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs* (3rd rev. and extended ed.). Berlin ; New York: Springer-Verlag.

[10] Wiegand, R. P. (2003). *An analysis of cooperative coevolutionary algorithms.* Ph.D. Dissertation, Department of Computer Science, George Mason University, Fairfax, VA, USA.

[11] Rosin, C. D., & Belew, R. K. (1996). *New methods for competitive coevolution* (Technical Report No. CS96-491). San Diego, CA: Department of Computer Science and Engineering, University of California.

[12] Maynard Smith, J. (1982). *Evolution and the theory of games*: Cambridge University Press.

[13] Axelrod, R. M. (1984). *The evolution of cooperation*. New York: Basic Books.

[14] Axelrod, R. M. (1987). Evolving new strategies: the evolution of strategies in the iterated prisoner's dilemma. In L. Davis (Ed.), *Genetic Algorithms and Simulated Annealing*. San Mateo, CA: Morgan Kaufmann Publishers, 32-41.

[15] Hillis, W. D. (1991). Co-evolving parasites improve simulated evolution as an optimization procedure. In C. G. Langton, C. Taylor, J. D. Farmer & S. Rasmussen (Eds.), *Artificial Life II* (Vol. X). Santa Fe Institute Studies in the Sciences of Complexity, New Mexico, USA: Addison-Wesley, 313-324.

[16] Paredis, J. (1994). *Co-evolutionary constraints satisfaction.* In Y. Davidor, H.-P. Schwefel & R. Männer (Eds.), Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN-III), Jerusalem, Israel, 46-55.

[17] Potter, M. A., & De Jong, K. A. (1994). *A cooperative coevolutionary approach to function optimization.* In Y. Davidor, H.-P. Schwefel & R. Männer (Eds.), Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN-III), Jerusalem, Israel, 249-257.

[18] Potter, M. A. (1997). *The design and analysis of a computational model of cooperative coevolution.* Ph.D. dissertation, Computer Science Department, George Mason University, Fairfax, VA.

[19] Maher, M. L., & Poon, J. (1996). Modelling design exploration as co-evolution. *Microcomputers in Civil Engineering, 11*(3), 195-210.

[20] Nair, P. B., & Keane, A. J. (2002). Coevolutionary architecture for distributed optimization of complex coupled systems. *AIAA Journal, 40*(7), 1434-1443.

[21] Shelton, K. (2005). *Development of robust design concepts using a co-evolutionary-based methodology.* Ph.D. Proposal, School of Information Technology & Engineering, George Mason University, Fairfax.

[22] von Neumann, J. (1951). The general and logical theory of automata. In L. A. Jeffress (Ed.), *Cerebral Mechanisms in Behavior: The Hixon Symposium*: Wiley, 1-32.

[23] Ilachinski, A. (2001). *Cellular automata: a discrete universe*. Singapore; River Edge, NJ: World Scientific.

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

[25] Kundu, S., Oda, J., & Koishi, T. (1997). *A self-organizing approach to optimization of structural plates using cellular automata.* In W. Gutkowski & Z. Mroz (Eds.), Proceedings of the Second World Congress on Structural and Multidisciplinary Optimization (WCSMO-2), Zakopane, Poland, 173-180.

[26] Kita, E., & Toyoda, T. (2000). Structural design using cellular automata. *Structural and Multidisciplinary Optimization, 19*, 64-73.

[27] Hajela, P., & Kim, B. (2001). On the use of energy minimization for CA based analysis in elasticity. *Structural and Multidisciplinary Optimization, 23*(1), 24-33.

[28] Altshuller, G. (1969). *Algorithm of invention*. Moscow: Moskowskij Raboczij Publishing House.

[29] Arciszewski, T., & Zlotin, B. (1998). *Ideation/Triz: Innovation key to competitive advantage and growth*, Internet, http://www.ideationtriz.com/report.html

[30] Terninko, J., Zusman, A., & Zlotin, B. (1998). *Systematic innovation, An introduction to TRIZ*: St. Lucie Press.

[31] Clarke, D. (1997). *TRIZ: through the eyes of an American TRIZ specialist*: Ideation International.

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

[33] Kicinger, R., Arciszewski, T., & De Jong, K. A. (2005). Evolutionary computation and structural design: A survey of the state of the art. *Computers & Structures*, in print.

[34] Bentley, P. J. (2000). *Exploring component-based representations.* In I. C. Parmee (Ed.), Proceedings of the Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM'2000), University of Plymouth, UK, 161-172.

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

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

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

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

[39] Kicinger, R., Arciszewski, T., & De Jong, K. A. (2005). Evolutionary design of steel structures in tall buildings. *Journal of Computing in Civil Engineering*, *19*(3), (in print).
[40] Kicinger, R., Arciszewski, T., & De Jong, K. A. (2004). Morphogenic evolutionary design: cellular automata representations in topological structural design. In I. C. Parmee (Ed.), *Adaptive Computing in Design and Manufacture VI*. London, UK: Springer-Verlag, 25-38.
[41] 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.
[42] Kicinger, R., Arciszewski, T., & De Jong, K. A. (2004). *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, 411-418.
[43] Kicinger, R., Arciszewski, T., & De Jong, K. A. (2005). *Generative design in structural engineering.* In Proceedings of the 2005 ASCE International Conference on Computing in Civil Engineering, Cancun, Mexico.