Genetic Algorithms in Architecture: a Necessity or a Trend?

Eleftheria Fasoulaki
Master of Science in Architecture, Computation Group
Department of Architecture, Massachusetts Institute of Technology
Email: efasarch@mit.edu

Abstract

Genetic Algorithms (GAs), a computational technique based on the principles of evolution, have been recently introduced in architecture to address problems of complexity in the function and the form of architectural projects. While there has been an increasing interest in the use of GAs in architecture, there has not yet been a systematic study of the operation of GAs and their application in architecture yet. This paper investigates whether the utilization of GAs is a necessity or a trend; whether GAs are used to accommodate specific needs of architecture or merely to bear innovative and complex forms; and consequently whether GAs serve reality or utopia. In order to answer these questions, the paper examines the operation of GAs in other disciplines and in architecture as well as the implications of its applications in architecture. Finally, the paper demonstrates the premises for a successful operation of GAs in architecture.

Introduction

Since the 1990's a shift has been noticed in the way avant-garde architects have used new technologies of evolutionary biology to address or depict the increased complexity that is noticed in today's architecture. Indeed, the layer of complexity that is introduced cannot be resolved by conventional design methods. Likewise, the quantity of information and the level of complexity involved in most building projects surpass designers' abilities to thoroughly comprehend and predict them. Genetic Algorithms (GAs), among many other evolutionary techniques, have been used in architecture as optimization tools or as form-generation tools. In the former, GAs address well-defined building problems, such as structural and mechanical. Genetic Algorithms are used as stochastic methods for solving optimization and search problems, operating on a population of possible solutions. Hereafter, this utilization of GAs is addressed as "necessity". In the later utilization, GAs are used under the scope of the concept of *emergence*. Genetic Algorithms are used to produce innovative representations and descriptions of processes by which emergent structures, often with tremendous complexity, are derived. Hereafter, this utilization of GAs is addressed as "trend".
While the interest in the use of GAs in architecture is increasing, no systematic investigation of the operation of GAs has yet been performed. The goal of this paper is to investigate whether the use of GAs is a necessity or a trend; whether GAs are used to accommodate specific needs of architecture or merely to bear innovative and complex forms; and consequently whether GAs are used to serve the architecture of a real world or abstract shapes of a conceptual world. To answer these questions I will examine how GAs are used in other disciplines compared to architecture so as to come to a conclusion whether architecture can adopt this process. If so, I will inquire whether GAs can be used in every phase of the

design process or if their characteristics constrain their utilization to a very specific phase. Finally, I will investigate whether GAs' utilization have changed the conventional design process or the role of architects.
This study is divided in three main parts. The first part examines the function, the operation and the applications of GAs in other disciplines. The second part investigates the dual utilization of GAs in architecture and analyzes two representative examples. The third part examines whether architects use GAs today due to necessity or due to trend, how the design process and the role of the designer changes, and through what criteria the final output is selected. Moreover, the third part promotes an alternative use of GAs in architecture: the combination of the both utilizations: the "necessity" and the "trend".

GAs in science

The introduction of computers into scientific and engineering fields has been one of the most revolutionary developments in the history of these disciplines, affecting many aspects of research. Most importantly, this revolution has helped humans explore, predict and control nature in ways that were inconceivable even fifty years ago.

Evolutionary Biology

Evolutionary biology studies the origin, the change and the multiplication of species over time. In evolutionary biology, the enormous set of possibilities of prospective genetic sequences, and the desired "solutions" are the results of highly fit organisms that are able to survive and reproduce within their environments. Due to that fact, evolutionary biology is an appealing source of inspiration for addressing complex computational problems that require searching through a huge number of possible solutions. Moreover, evolution can be seen as a massively parallel search method; rather than work on one species at a time, evolution tests and changes whole populations of species simultaneously. The natural procedures of life evolution and the techniques that are used in evolutionary biology have influenced many other disciplines that use evolutionary algorithms to solve complicated problems. A particular class of these computational algorithms is Genetic Algorithms.

Genetic Algorithms

Genetic Algorithms were invented by John Holland in the 1960s and since then they have been used as stochastic methods for solving optimization and search problems, operating on a population of possible solutions. According to Darwin's Theory of Evolution, the repetitive application of the aforementioned procedures alters an initial species into various other species; however, only the stronger prevail. Genetic Algorithms perform the same operations on the population of possible targets with only those that fit the solution better surviving.
Even though there is no formal definition of GAs, all of them consist of four elements. The first is the populatio*n* of chromosomes which represent the possible solutions of the problem. Selection is the second element and it refers to the part of the population that will evolve to the next generation. Selection is performed based on a fitness function, that determines how "good" a solution is. The selection process is applied to each generation produced. Crossover refers to the combination or exchange of characteristics between two members of the elite group defined by selection, by which offspring is produced. There are various types of crossover but the most frequently used are: the one-point crossover, in which the parents are cut at a specific point and the head of the first is pasted to the tail of the second or vice versa;

and the two-point crossover, in which a part from one of the parents is obtained and exchanged with the part that lies in the same location of the other parent.

Parent 1	110 / 0100110	Parent 1	110 / 0100 / 110
Parent 2	101 / 1010101	Parent 2	101 / 1010 / 101
Offspring 1	110 1010101	Offspring 1	110 1010 110
Offspring 2	101 0100110	Offspring 2	101 0100 101

Table 1 - One and Two points Crossover

After the application of crossover on the population, a new generation is produced. Whether parents are part of the new generation or not is an option that depends on the problem. In any case, before re-applying selection to the new population, mutation takes place. Mutation is a random event, occurring with a user-defined probability to only some of the new offspring. It is used to maintain genetic diversity by altering only a little piece of the new offspring.

Parent 1	110 / 0100110
Parent 2	101 / 1010101
Offspring 1	110 1010101
Offspring 2 *(mutated on the 1st bit)*	101 0100110

Table 2 Mutation

All the methods described above rely heavily on the nature of the problem to be solved, the domain in which the solutions are to be found, and the encoding of the solutions. More complex encoding structures, such as digital trees, allow more difficult problems to be solved, but also require more complex methods to be defined for the manipulation of the generations. However, the basic structure of the GAs remains the same and is outlined below.

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness $f(x)$ of each chromosome x in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete

3.1. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)

3.2. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.

3.3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).

3.4. **[Accepting]** Place new offspring in a new population

4. **[Replace]** Use new generated population for a further run of algorithm
5. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population
6. **[Loop]** Go to step **2**

Table 3 Outline of the Basic Genetic Algorithm

Applications of GAs

Like other computational systems inspired by natural systems, GAs have been used in two ways: as techniques to solve technological problems and as simplified scientific models that can answer questions about nature. Genetic Algorithms address quite a large number of problems including image processing, face recognition, protein structure prediction, time series analysis, computer software automatic evolution, cellular automaton rule evolution, robotics, control, aeronautics, and many more. Fields in which GAs have been extensively used include Optimization, Automatic Programming, Machine Learning, Economics, Immune Systems, Ecology Population Genetics, Evolution and Learning, and Social Systems.

These lists are by no means complete, but illustrate the variety of applications that GAs offer in various fields.

GAs in Architecture

While other disciplines have adopted computational tools based on the principles of evolutionary biology, in architectural design evolutionary processes have not been broadly applied. Only recently has there been a noticeable shift in the way architects explore such techniques to address complex problems. Indeed, one of the main problems in architecture today is the quantity of information and the level of complexity involved in most building projects.

Genetic Algorithms offer an effective solution to this problem by solving optimization and search problems, operating on a population of possible solutions. In architecture GAs operate in two ways: as optimization tools and as form-generation tools. In the first way GAs address well-defined building problems, such as structural, mechanical, and thermal and lighting performance. In the second way GAs are used under the scope of the concept of *Emergence*. The dual operation of GAs in architecture will be analyzed hereafter.

GAs and Design Optimization

Design optimization has been introduced to building industry as a tool to achieve the best possible building performance, the highest reliability and / or the lowest cost. Building performance includes among others the structural, acoustic, lighting, energy and spatial attributes/properties of a building. For example one of the basic aims of structural optimization is to minimize the overall weight so as to minimize the material cost. With the increased demands of the global market for more effective and complex buildings, the utilization of GAs, as one of numerous optimization techniques, is a necessity. Especially for large-scale structures with thousands of elements or structures with very complicated geometry manual calculations can not satisfy the increased demand so the use of optimization techniques is inevitable. For example, in the project for the Aquatics Centre for the 2008 Olympic Games in Beijing, the new automated approach for selecting section sizes and checking them to design codes for all 25 000 steel sections was crucial for the feasibility of the project's roof.

Most optimization problems are made up of three basic components. The first is the objective function which we want to minimize or maximize, the second component is the designation of a set of design variables that affect the value of the objective function, and the third component is the determination of a set of constraints that allow the design variables to have certain values. For instance, in terms of the structural performance of a panel, we determine what we want to minimize that might be the stress in a particular region, and then we determine the variables, that could be the geometry and material of the panel, and then we set the constraints that could be the minimization of the weight of the panel.

For a thorough investigation of the operation of GAs as an optimization technique in design the Genetic Algorithm Tool for Design Optimization implemented by Luisa Caldas and Leslie Norford in 1999 will be examined. The Genetic Algorithm Tool [3] explores the use of GAs in the context of generative and goal-oriented design in order to develop and evaluate criteria related to the environmental performance of a building. The tool searches for the optimal window size in a building in order to optimize three characteristics of the adjacent room: lighting, heating and cooling performance. Environmental conditions including climate conditions, window orientation, and glazing, are parameterized since they influence the achievement of low-energy consumption solutions.

Two office buildings are examined, both with controlled internal climate and artificial lighting. The buildings are located in two different cities: Phoenix, Arizona and Chicago, Illinois. After the GAs have generated possible design solutions, the designs are then evaluated in terms of lighting and thermal performance through a detailed thermal analysis program (DOE2.1 E). Then the GAs use the results from these simulations to further investigate towards finding low-energy solutions to the problem under study. Solutions are visualized using an AutoLisp routine, since AutoLisp procedures allow the results to be visually inspected as AutoCAD drawings.

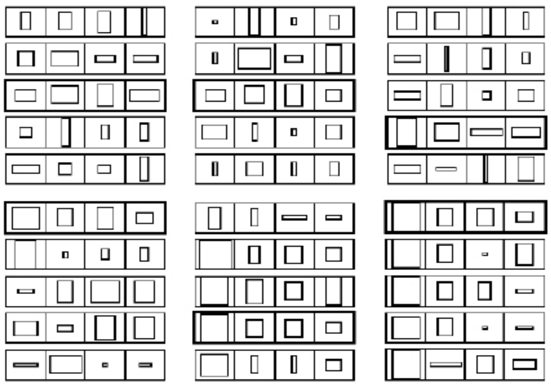


Figure 1 Generations for Phoenix project
hhtp://cumincad.scix.net/sci-bin/works/Show?b5d2

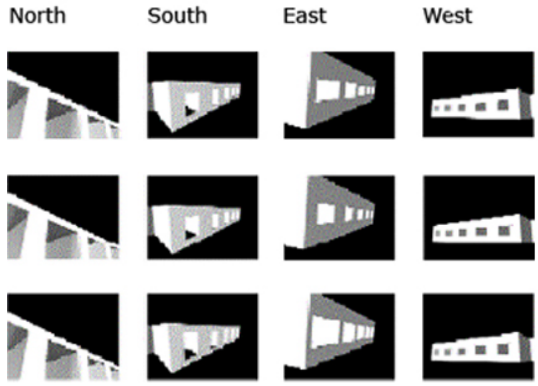


Figure 2 Final solutions for Phoenix project

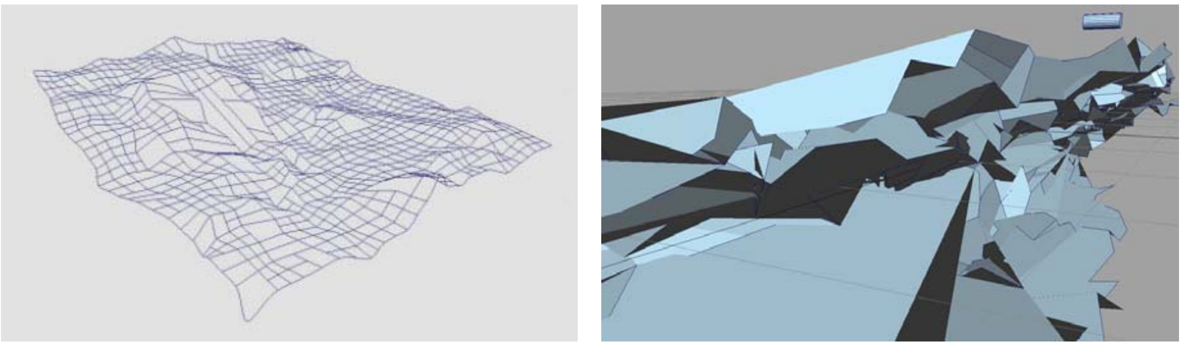
GAs and Emergence

Emergence was introduced as a subset of complexity theory in the 1980s and it is linked back to the development of systems theory in the 1920s. Emergence refers to the universal way in which small parts of systems in nature driven by very simple behaviors are tended toward coherent organizations with their own distinctly different behaviors. Vivid examples from the natural world are the hive, swarming, and flocking where independent parts are formed into one system with a complex or / and random behavior.

While such models are used to generate novelty design or evolving forms, only recently have architects started to explore the ideas of Emergence and examined the new technologies in the natural science to enrich the scopes of architecture, redefining new formal paradigms. Avant-garde architects have referred to theoreticians such as Rene Thom and D'arcy Thomson to investigate the ways in which biological ideas can operate and be generative in architecture. The term of Emergence equates architecture with nature assuming that design is dominated by the same principles as the natural world. Architects attempt to create architecture as nature; architecture that is nature. That is why the notion of emergence is linked tightly with the notion of growth, evolution, continuity and behavior. Indeed, behavior is a dynamic process of feedback between states of forming no axis or center. However, if architects want to use this concept, they have to equate buildings with the living organisms, parts with the whole and function with behavior. In this case, each unit is a part of an environment defined by the building and its neighbors. To produce such environments, architecture must be expressed as rules of generation, in order to allow evolution to be defined.

In this context GAs are introduced, offering a bottom-up approach in multiple cycles of evolution. For a thorough investigation of the operation of GAs as a form-generation technique the *Generative Form Modeling and Manufacturing (Genr8)* [4] developed by the Emergent Design Group in 2001 will be examined. The Generative Form Modeling and Manufacturing

is a tool that combines a generative design algorithm -- Map L-systems -- with evolutionary search algorithms -- GAs. Through the combined use of these two algorithms, Genr8 designs the geometry of surfaces and allows the user to interfere with growth, by stopping evolution, altering fitness and run-time parameters, including the environment in which growth takes place. Genr8 was used to conduct an experiment to explore the potential of achieving a coherent coexistence of the logic behind manufacturing, constraints implied by construction materials and complex geometrical forms.
The development of the surfaces is an outcome of two processes: definition and re-writing rules. The application of these rules to the initial definition and any subsequent development alters all parts of the surface, and the results are continuously translated into three dimensional drawings. The conditions that influence the growth of the surface also influence the grammar of the GAs and along with innate factors define the form of the surface. Growth is achieved with the use of the GAs. A critical issue in the use of the GA is the definition of the fitness criteria. These are defined as mathematical functions representing properties of the form of the surface including size, smoothness, soft boundaries, subdivisions, and symmetry.



Figures 2-3 Images generated by Genr8, http://mit.edu/edgsrc/www/genr8/media.html

GAs: Necessity or/and Trend

GAs in architecture and other disciplines

Based on the former analysis, we can argue that there is a basic difference between architecture and other disciplines in the utilization of GAs. Unlike other fields that address problems whose targets are well-defined, many of the problems that architecture deals with are ill-defined. Liddament, in his article “The Computationalist Paradigm in Design Research,” [5] shows that although computational tools, such as GAs are powerful in scientific domains to solve many problems, they do not adequately fit the actual design activity. He acknowledges the fact that the computationalist paradigm presents itself as a “scientific approach with a correspondingly rigorous methodology.” Design intention, for example, is usually an ill-defined problem. Even if designers finally manage to code design intention, this process will not be enough to guarantee a successful design solution. This is because an architectural project is a composition of design performances including spatial, structural, lighting, acoustic, and thermal. These elements continuously interact during the design process. Consequently, the optimum solution of one of those elements does not lead to a successful combination of all elements, hence, to a global optimum and successful design solution.

Another noticeable difference between architecture and other fields is that architecture is not just a rule-driven science. While architecture primarily serves the function of the buildings, it also deals with cultural, social and aesthetical notions whose codification and fitness is subjective, but still could be encoded into rules.

GAs: Necessity or Trend?

Despite the difficulties in coding many architectural problems, algorithmic processing of GAs -- as a way that builds human's thought to solve problems -- can be used in the design process. Taking this into account, the next question that is addressed is whether the use of GAs in architecture today is driven by necessity or trend.

Necessity

The increased human needs and the today's lifestyle call for more complicated functional requirements and the quest for more innovative forms augments the complexity of the formal manifestation. The level of complexity that is introduced and the quantity of information that it entails constitute one of the basic problems that architecture must deal with today. This problem cannot be resolved by conventional design methods. Likewise, the constraints of this problem surpass designers' abilities to thoroughly comprehend them and predict their solution. Seeing GAs as tools that can answer to the specific needs of practical architecture and not merely of experimental design, the notion of *necessity* of GAs arises.
Genetic Algorithms, as stochastic processes for solving optimization and search problems, go through thousands of iterations in a second and find the solution sets, extending designers' thoughts into a once unknown and unimagined world of complexity. Under the scope of *necessity* architectural projects utilize GAs as optimization tools to address well-defined problems, such as the structural, mechanical, thermal and lighting performance of a building. In this case, GAs serve design for architecture, design for the real world. This is the reason why this utilization of GAs is called a *necessity*.
However, even in the case of necessity there is a possibility for architects to select GAs -- among numerous other optimization techniques -- just because GAs are a cutting edge method. Indeed, there are many other optimization methods, including hill climbing, simulated annealing, tabu search, stochastic tunneling, and harmony search. The selection of a particular optimization method should be based on the specific needs and nature of a problem and how efficiently the method can respond to these demands and not how trendy a method is. For example Prof. Kristina Shea used the method Structural Topology and Shape Annealing (STSA) to develop EifForm, a stochastic optimization software demonstrator for generative structural design. [6]

Trend

On the other hand, we have noticed a more generalized utilization of GAs as form generation tools. Influenced by the concept of emergence and evolutionary architecture, architects use GAs as means of exploring innovative forms. The number of architects who adopt these techniques is rapidly increasing. Likewise, their designs are getting more and more complex. Reading the description and observing the final output of those experiments, one could argue that formal complexity is the primary consideration for those architects. The fact that these forms often do not follow or serve any functional or structural requirement boosts this argument. At this point, a sequence of questions is raised:

Why should the complexity of buildings be increased today if it is not to justify functional or structural requirements?

Do those architects believe that only through complexity they will achieve formal innovation?

Do they believe that the formal complexity of buildings can reflect the complexity of the building environment and everyday life?

Why is formal complexity their primary task anyway?

The fact that those architects utilize GAs merely as form-generation tools, omitting spatial and structural performance, in conjunction with the fact that most of those projects do not produce architecture but rather abstract shapes which are difficult to be translated into architecture could lead to the conclusion that in this case, GAs serve design for abstract shapes, design for a conceptual world. This is the reason why this utilization of GAs is called a *trend*.

Also most of these architects do really respect the notion of Emergence or they have not truly understood the real meaning of this notion. Emergence is not interested in parts; it is the science of wholes. And since design process is comprised of many parts they should not omit them and focus only in formal representation. However, even if architects are able to combine many building requirements and performances with formal generation, a fallacy still exists in this attempt. This fallacy is based on the equalization of architecture to biology. Indeed there are some similarities between architecture and biology, both are materially and organizationally based, both are concerned with morphology and structuring. Nevertheless, those similarities do not lead to the perception of building as artificial life form, which is dominated by the same principles that the natural world is dominated. This speculation equates cultural evolution with organic evolution, which is the same as to equate the Darwinian with Lamarckian theories of evolution. Since form in architecture is a cultural artifact, imaging numerous abstract meanings as previously mentioned, form can not be subjected to the Darwinian definition of evolution. Architects must clarify where architecture is literally considered as part of nature, where there are analogies or metaphors, and where nature is a source of inspiration.

There is no doubt that abstract shapes of a conceptual world may be implemented in the future, opening the path for new formal expressions. However, the real challenge would be to use GAs in real-world architectural contexts. Indeed, it is very difficult for a designer to combine complexity and constraints imposed by the design problem with evolutionary formal generation. Possibly this is the reason why architects use GAs either as a tool to solve structural and mechanical problems or as a tool to generate forms.

Design process and designer

The changes in the design process and the role of design are one of the most important implications of the utilization of GAs. In the case where GAs are used as an optimization method those changes are not that important. Even if at first glance it may appear that those methods lead to the replacement of the designer from the design loop, this is not really the case. Indeed, the designer or the engineer is the one who decides what will be optimized setting the variables and the functions of the problem while the optimization techniques are just another tool available to the designers shorting the design cycle times and performing the calculations.

On the other hand, when GAs are used as a form generation tool these implications are more important, since GAs are used during the conceptual design phase. Evolutionary simulations replace the traditional design processes and the designer in a sense is neutralized and marginalized. This is because most of the designers use GAs to breed new forms rather than merely design them. As fascinating as the idea of breeding buildings inside a computer may be for some designers, it is clear that merely using digital technology without functional, structural and topological thinking will never be adequate for real architecture. Consequently, the main role of the designer is to be the judge of aesthetic fitness. In the words of Steadman: "Just as Darwin inverted the argument from design, and 'stole away' God as designer, to replace Him with natural selection, so the Darwinian analogy in technical evolution removes the human designer and replaces him with the 'selective forces' in the 'functional environment' of the designed object." [7]

Selection

The selection of the final output is another issue that is derived by the use of GAs. In the case of *necessity* the target is well-defined and consequently the selection of the result is based on the optimal solution. For example, in the project of Caldas and Norford that was previously examined the target was the optimal window size, in terms of lighting, heating and cooling performance. Based on this, the final output the sizes of the windows were selected based on the lowest-energy consumption. However, in the case of *trend* the target is ill-defined and the criteria for the selection are only aesthetic. At this point a sequence of question is raised:

> Why should designers follow such a strict, logical, and subjective process, if they eventually base their selection on abstract and objective aesthetic criteria?
> Why do they not design something similar to the final output at the very beginning?
> Is it because some architects merely use GAs as a tool to surpass their limitations of creativity?
> Is it because they believe that the utilization of GAs document their design process, providing it with a "theoretical" and conceptual substance?

GAs: Necessity and Trend

Undoubtedly there are many difficulties in the application of GAs in architecture. However, GAs have the potential to play a more effective role in the future of architecture. On one side it is the architectural problem that has a "reason" and a fundamental "factor." The "reason" is the increasing quantity of information and the increasing level of *complexity* involved in most building projects today. The "factor" is the various building *performances* including spatial, functional, aesthetic, structural, energy, and lighting. All these performances interact and interlink and can not be considered separately. On the other side there is the potential of GAs. Genetic Algorithms are search methods for addressing *complex* problems, finding optimum design solutions from indeterminate search spaces constrained by *multiple input* factors. If architects want to use computational design to address architectural problems Generative Algorithms can be one of possible tools to do so; they can solve the "reason", taking into account the primary "factor".

Yet, in order for GAs to be applied in architecture, some things must be done. Today the usage of GAs is local; that means that GAs are used merely for form generation or for optimization of one building performance. The problem in today's utilization is that the local function of GAs is influenced during the design process and, in the end, the local "optimal" is lost. For example, if a designer utilizes GAs so as the form of a building to be emerged, in the next step of the design process he/she will need to change the form partly or radically so as to fit the function. The same will happen if a designer uses GAs to find the optimal form minimizing the cost and maximizing daylight. Then, if he/she tries to calculate the thermal performance with the conventional tools the design will risk loosing the local optimal which

has been calculated in the first step, since in the next step of the process another performance must be calculated and new functions will be introduced.
A possible solution to this problem is the coordination of generative tools with optimization tools; the combination of dual utilization of GAs: trend and necessity. The form will emerge by the simultaneous calculations of most of the performance-driven functions. This process can be seen as a potential effort to achieve the design goal finding the 'fit' between form and context, defining the context as "anything in the world that makes demands of the form" [8] -- including meanings, aesthetics, environment, and function -- as Christopher Alexander stated. Indeed, for Alexander "the form is the solution to the problem; the context defines the problem." [9]. This process will help architects to find global "optimal" or satisfactory solutions, aid multi-disciplinary negotiations, shorten the project delivery time and cost, and find feasible design alternatives. However, some potential negative consequences will be derived by the application of this process. Architects must make sacrifices related to the generalizations and reductionisms of some of the design problems. Those must be done in order to transform most of the ill-defined problems -- that are related to the subconscious and subjectivity of the architect, such as the design intention -- into well-defined ones, by coding them. Undoubtedly, this transformation will be very difficult to achieve. Many authors have referred to the difficulties of coding the very complex and unexpected way that the human mind functions. On the other hand these transformations seem inevitable since the constraints of recent architectural problems surpass designers' abilities to thoroughly comprehend them and predict their solution. Another aspect that architects must take into account applying this method is the fact that many problems emerge or diversify during the design procedure. Consequently, the interactivity between designer and computational tool must be accommodated allowing the designer to add/reduce the variables or changing the fitness functions.

Conclusion

One of the problems that contemporary architecture has to deal with is the quantity of information and the increasing complexity of most of the architectural projects. Only recently have architects started to utilize Genetic Algorithms (GAs) to address this problem. This paper demonstrated the dual operation of GAs in architecture: as optimization tools, and as form-generation tools, addressing these as necessity and as trend respectively. This paper also indicated the implications of GAs' applications. Some of these implications include the replacement of the traditional design process by the evolutionary simulations, the neutralization and rescission of designers, the abstract criteria of the final selection, and the local utilization of GAs only in some of the design phases. However, GAs have the potential to play a more effective role in the future of architecture. Indeed, GAs can answer the architectural problem taking into account the processes of the design, if they are properly utilized. A possible solution to this problem is the coordination of generative tools with optimization tools so as to achieve a simultaneous calculation of the performances and a global evaluation of the design which will be emerged by the performance-driven functions.
The methodology of incorporating a number of search spaces has been applied in other engineering fields, such as aeronautics and astronautics, leading for example to lighter, stronger, stiffer, and often cheaper automotive bodies, airplane wings, and ship keels. This method is called Multidisciplinary Design Optimization (MDO). Through the utilization of GAs and other evolutionary techniques, MDO solves complex coupled systems, exploring the interacting disciplines or phenomena at every stage of the design process. Further research in

the application of GAs will involve investigating the operation and possible application of MDO in architecture.

References

[1] William Lethaby, Architecture: An Introduction to the History of the Art of Building, (T. Butterworth Ltd; Rev: 1911), 25.

[2] Micheal Hensel, Achim Merges, “Emergence in Architecture”, Architectural Design 74, no 3 (May/June 2004), 6-9.

[3] Luisa Caldas, Leslie Norford, A genetic Algorithm Tool for Design Optimization, Media and Design Process [ACADIA ‘99] Salt Lake City, (29-31 October 1999): 260-271, http://cumincad.scix.net/cgi-bin/works/Show?b4d2, retrieved November 01, 2007.

[4] Peter Testa and Devyn Weiser, “Emergent Structural Morphology”, Architectural Design 72, no1, (January 2002), 13-16.

[5] T. Liddament., “The Computationalist Paradigm in Design Research.” *Design Studies,* Vol 20, No 1, (1999) 44.

[6] Kristina Shea, Robert Aish, Marina Gourtovaia, “Towards integrated performance-driven generative design tools” [eCAADe ‘03] Graz University of Technology (Austria): 553-560, http://iam.tugraz.at/~e/cd/ecaade_ files/pdfs/Friday/149_shea.pdf, retrieved November 01, 2007.

[7] Philip Steadman, The Evolution of Designs: Biological Analogy in Architecture and the Applied Arts. (New York, Cambridge University Press: 1979), 189.

[8] Christopher Alexander, Notes on the Synthesis of Form. (Cambridge, Harvard University Press: 1964), 15.

[9] Ibid.