The Application of Evolutionary, Generative, and Hybrid Approaches in Architecture Design Optimization

Hoda-Esmaeilian Toussi
Department of Architecture, Faculty of Architecture, NEU University, hoda.esmaelian@neu.edu.tr

Abstract

Since the emergence and application of evolutionary optimization approaches in architecture in the early twentieth century, a wide range of studies have attempted to integrate evolutionary strategies with the design process. The extensiveness and dispersion of research in this field and the growing application of the generative evolutionary techniques in solving design problems necessitate analytical classification of pertinent literature review. Based on the descriptive-analytical review of the literature on generative evolutionary strategies in architecture, this paper proposes a research model for an integrated generative design framework to enhance the future application of this approach in the conceptual design stage. Therefore, first, selected 140 journal articles, with key-word exploration method, between 2014 and 2020 is analyzed to categorize the applied techniques, identify the gap, and address the issue of selecting the appropriate evolutionary approach in the early stage of design. Literature analysis is classified into seven topics, each demonstrating shortcomings of related studies in four categories of form finding, Spatial Programming, Performance-based optimization, and Multi-objective optimization. The research results indicate a growing interest in applying hybrid methods, multi-objective optimization problems, the need for an integrative generative evolutionary framework in the early design phase, and a conceptual design tool with Co-simulation possibility.

Keywords: evolutionary optimization algorithm, multi-objective optimization, generative design, hybrid methods

Mimari Tasarım Optimizasyonunda Evrimsel, Üretken Ve Hibrit Yaklaşımların Uygulanması

Özet

Yirminci yüzyılın başlarında mimaride evrimsel optimizasyon yaklaşımlarının ortaya çıkması ve uygulanmasından bu yana, geniş bir çalışma yelpazesi, evrimsel stratejileri tasarım süreciyle bütünleştirmeye çalışmıştır. Bu alandaki araştırmanın yaygınlığı ve dağılımı ve tasarım sorunlarının çözümünde üretici evrimsel tekniklerin artan uygulaması, ilgili literatür taramasının analitik sınıflandırmasını gerektirmektedir. Mimaride üretken evrim stratejileri ile ilgili literatürün tanımlayıcı-analitik incelemesine dayalı olarak, bu makale kavramsal tasarım aşamasında bu yaklaşımın gelecekteki uygulamasını geliştirmek bağlamında entegre bir üretken tasarım çerçevesi oluşturmak için bir araştırma modeli önermektedir. Bu nedenle, öncelikle 2014 ve 2020 yılları arasında ve anahtar kelime aracılığıyla seçilen 140 makale, uygulanan teknikleri kategorize etmek, boşlukları belirlemek ve tasarımın erken aşamasında uygun evrimsel yaklaşımın seçilmesi konusunu ele almak için analiz edilmiştir. Literatür analizi, her biri dört kategoride, Mekansal Programlama, Performansa dayalı optimizasyon ve Çok amaçlı optimizasyonda form bulma ilgili çalışmaların eksikliklerini gösteren yedi konuya ayrılmıştır. Araştırma sonuçları, hibrit yöntemlerin, çok amaçlı optimizasyon problemlerinin uygulanmasına olan ilginin arttığını ve erken tasarım aşamasında bütünleştirici bir üretken evrimsel çerçeveye ve eş zamanlı simülasyon olasılığına sahip kavramsal bir tasarım aracına duyulan ihtiyacı göstermektedir.

Anahtar Kelimeler: evrimsel optimizasyon algoritması, çok amaçlı optimizasyon, üretken tasarım, hibrit yöntemler

INTRODUCTION

Design process encompasses iterative activities of data collection, problem definition and exploration, ideation and evaluation. Due to extensive, complex, and seemingly contradictory involved design aspects, a shift towards generative and evolutionary design in the field of architecture has occurred which replicates natural evolution process in the virtual spaces of the computer (Frazer J. H., 2002). This biology-driven approach has provided designers with a diversified search space and design options as well as obtaining specific goals (HM., 2006).

Since 1970, with the growing application of evolutionary techniques in form finding and solving multi-objective optimization problems in architecture, extensive studies have been conducted in response to proposing a comprehensive generative evolutionary framework in the early design phase. The broad range of research can be exemplified in studies related to form finding (Ceccato, 1999), (Kicinger, 2005), (Janssen, 2005), engineering research field (Rosenman, 1999), (J., 1995), (Poon J., 1996), evolutionary algorithm development (Gong, 2008), (Mühlenbein, 1993), design notion (HM., 2006), (Gu, 2006), multi-objective optimization (Lee L. H., 2008), (Limbourg, 2008), and hybrid systems (Nariman-Zadeh, 2005), (Park, 2007). Aiming to identify a comprehensive generative evolutionary approach, this paper applies a descriptive literature review to select, analyse, and categorize the related literature among journal articles from 2014 to 2020. The first section of the paper defines research area to help clarify potential keyword exploration. The second part examines the selected articles based on four design objectives of "form finding", "performance-based optimization", "Spatial Programming", and "multi-objective problems". This classification facilitates the literature analysis process and demonstrates the number of studies with only one specific goal as opposed to adopting a multi-objective outlook in the early design stage. Also, the efficacy of various applied evolutionary techniques in different articles can be compared more precisely when they share a common objective.

RESEARCH METHODOLOGY

In this article a descriptive literature review was adopted to identify interpretable patterns and gaps in academic published journals, using qualitative descriptive statistics with info graphic representations. This method includes seven different stages as is illustrated in Figure 1. The first step, initialization, uses library research to identify and define related keywords. The second step is to search designated academic database, naming Scopus, Science Direct, and Google Scholar for review journal articles, with preferably high citation, between 2014 and 2020, English language, and based on keywords "evolutionary", "generative", "multi-objective", "optimization", "conceptual", "design" with excluding "urban" and "town", as urban design is not in the scope of this research. The third step involves sorting and organizing data based on publication year, citation, relevance to design and architecture field, and the use of case study. Among 250 journal articles extracted from the second step research, 100 relevant articles were selected in the fourth step. Also, to extend the search space the keywords "Shape Grammar", "form finding", and "Spatial Programming" were added to the step 2 keywords one by one, and step 2, 3, and 4 were repeated to obtain the pertinent data. In the fifth step, the studies were classified based on the main objective of the research containing, "form finding", "performance-based optimization", "Spatial Programming", and "multi-objective problems" for better evaluation of techniques and tools. Each of the documents was analysed based on the applied technique, algorithm, and generalization process in the sixth step. Finally, based on the achieved analysis conclusion, a research model for applying the integrative generative evolutionary framework was proposed.

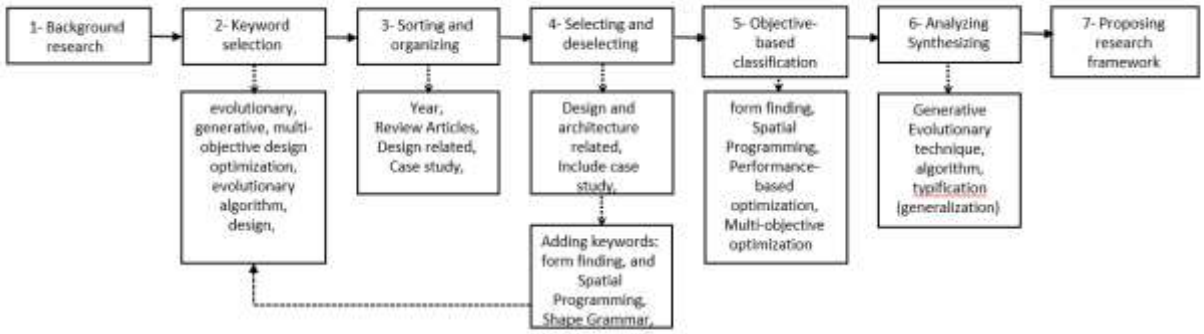

Figure 1: Seven-step research structure. Source: Author

BACKGROUND RESEARCH

Evolutionary Computation (EC) and Evolutionary Design

Design is "*a dynamic process of adaptation and transformation of the knowledge of prior experiences in order to accommodate them to the contingencies of the present* (Oxman, 1990)". In essence, the design process is a recursive procedure, attempting to reconcile complex, contradictory involved elements which can consequently be defined as a continuous problem definition process. To address and support this level of complexity, Evolutionary Computation has been applied and examined by many scholars)Sims, 1994(, (Frazer J. , 1995), (Bentley, 1999). According to Bentley (p., 1999), Evolutionary Computation and Evolutionary Design are rooted in Computer Science and Evolutionary Biology. This field has emerged as an extensive approach by scholars such as Holland (Holland, 1975), Rechenberg (Rechenberg, 1978), and Fogel (Fogel, 1963) to integrate Evolutionary Biology and Computer Science (p., 1999). Evolutionary Computation (EC) is based on the evolutionary biology, mimicking the natural evolution of real life. Explaining briefly, natural evolutionary process can be defined as a series of activities, including selection, crossover, and mutation, that lead to the spread of inherited traits, increasing the probability of survival and reproduction of an individual in a population over successive generations. The fitness of these individuals depends on the environment and their potentials in achieving their goals (Eiben, 2015). Evolutionary Computation, inspired involves Evolutionary Algorithm (EA), Evolutionary Strategy (ES), Genetic Algorithm (GA), and Genetic programming (GP), which all mimic the natural evolution, albeit with some differences in the mechanisms of mutation and crossover (Chan K.H., 2002). Table 1 depicts some of the evolutionary algorithm definitions and history.

Table 1: Evolutionary algorithm types, definition, and history

EVOLUTIONARY ALGORITHM	GENERAL DEFINITION	ORIGIN
GENETIC ALGORITHM (GA)	*Genetic Algorithm is a family of computational models based on principles of evolution and natural selection. These algorithms convert the problem in a specific domain into a model by using a chromosome-like data structure and evolve the chromosomes using selection, recombination, and mutation operators* (Li, 2004)	Holland, 1962, in Ann Arbor, USA
GENETIC PROGRAMMING (GP)	*The automated process of improving system behaviour for solving non-linear problems using evolutionary algorithms* (Wang, 2016).	Fogel, 1962, in USA

EVOLUTIONARY STRATEGIES (ES)	*A set of rules for the automatic design and analysis of consecutive experiments with stepwise variable adjustments driving a suitably flexible object/system. Like with all classical methods, the performance of the evolution strategies largely depends on the adjustment of the internal parameters, prominently the mutation strength(s)* (Beyer, 2002)	Rechenberg, 1965, in Germany
DIFFERENTIAL EVOLUTION (DE)	*Differential Evolution (DE) is a parallel direct search method which utilizes NP D-dimensional parameter vectors (Storn, 1997). The population-based intelligent optimization algorithm is for solving continuous and discrete problems. The evolution process in this algorithm is based on gradual and continuous improvement in the candidate response and according to the principles of all evolutionary algorithms, it needs a fitness function* (Ho-Huu, 2016).	Storn and Price 1997,

John Frazer (1995) was among the first scholars to use evolutionary methods in design, especially in architecture and structural design, and to study the generative aspect of evolutionary algorithms (Frazer J. , 1995). Also, Karl Sims)Sims, 1994(reviewed early experiments of applying GA in graphic and virtual creature design. Various studies and projects have been carried out in relation to form finding for architectural and structural design with evolutionary processes (Kicinger, 2005), (Janssen, 2005). While this innovative approach can be applied for generating, evaluating, and exploring design problems' solutions, most of current related studies concentrate on one of these aspects, and mainly on the performance-based optimization process at the detailed design stage. Optimization is an important, decisive activity in design. Designers will be able to produce better solutions when they can save time and money with optimization methods. *Optimization is the process of adjusting the inputs to or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output or result* (Haupt, 2004). Generative and evolutionary methods have proven to be strong synergists for design exploration, and design optimization has been proposed as a method to assist the exploration process. Rarely is optimization intended to achieve an optimal solution, instead providing designers with insight into the solution space (Stouffs, 2015). Involving elements in this process contain: 1) Objective function (differs in various fields, naming Model Economic, Profit Function, Cost Function, Index performance and so on), 2) Constraint (defining system behaviour with functions and variables), 3) Decision Variables (which in optimization we seek to determine their values to achieve the optimal function), and 4) The type of variables that include Mixed Integer Programming. Optimization problems can be divided into 1- single-objective optimization problems and 2- multi-objective optimization problems based on the number of objective functions. In single-objective optimization problems, the goal of solving the problem is to improve a single objective function whose minimum or maximum value fully reflects the quality of the response obtained. But in some cases, especially in design problems, it is not possible to score a hypothetical answer to an optimization problem based solely on one goal. In this type of problem, we have to define several objective functions and optimize the value of all of them at the same time. Multi-objective optimization is one of the most active and widely used research fields among optimization topics. In architectural problems two types of issues can be mentioned, one is the existence of several conflicting goals, and the other is a very complex and extensive search space. Therefore, many studies have attempted to apply Multi-objective Optimization techniques to assess and obtain optimal solutions.

Generative Design System

Generative design can be defined as the process in which multiple potential solutions are identified by algorithms. Generative architecture is defined more generally by the use of a generative system, such as a set of language rules, a computer program, a set of geometric transformations, a diagram, or other procedural innovations in the design process by which the final design is produced. The generating system has different degrees of automation from fully automated process to step by step user-controlled process. This process includes designing the algorithm (rules), setting the initial shapes and parameters, advancing the adaptation process, and finally selecting the best option. The maturity of generative systems in architecture occurred after the development of architecturally-based software in the mid-twentieth century. One of the first systems written in architecture based on shape grammar (shape rules) was to generate Villa Paladin. Stiny and Mitchell (Stiny, 1978) created parametric shape grammar which not only generated Villa Paladin ground plans, but it also created novel ground plans coherent to their initial pattern. Their first attempt was to redesign parts of Palladio's architectural rules in a modern way and generate a form (Stiny, 1978). Table 2 defines some of common generative systems.

Table 2: Common generative design systems

GENERATIVE METHOD	GENERAL DEFINITION	ORIGIN
SHAPE GRAMMAR	George Stiny defines a shape grammar as 'a set of transformation rules applied recursively to an initial form, generating new forms' (Strobbe, 2015)	Stiny & Gips, 1972
CELLULAR AUTOMATA	Cellular automata (CA) are discrete models of space and time and typically involve interactions of cells across homogeneous lattice grids. Cells can take on a given finite number of cell states, which can change according to simple rules each cell executes in relation to its cell neighborhood (Herr, 2016)	John von Neumann, 1950s
LINDENMAYER SYSTEMS	L-System is an algorithmic digital generator which is based on the parallel rewriting system, a type of formal grammar, that can potentially produce natural fractals. Developed by a Hungarian biologist Aristid Lindenmayer in 1968, L-systems can reproduce the dynamic of plant growth, offering architects to apply this system of form generation in architectural designs (Rian, 2014).	Aristid Lindenmayer, 1968

Hybrid Systems

The approaches proposed so far, in the early stages of design, concentrate more on general optimization, thus, using innovative and meta-heuristic algorithm to replicate the simulation model, which often results in local optimum. In an effort to improve the efficiency of evolutionary methods to solve optimization problems, researchers have used a combined method. The hybridization process is done in the following ways:

1) Using an algorithm to create a population and then applying another method to improve the created population. 2- Using multiple parameters in an evolutionary method, and 3- Using local exploration to improve the solutions obtained from multi-objective optimization evolutionary methods (Thangaraj, 2011).

One of the most widely used hybrid methods is the simultaneous application of several similar algorithms with different parameters which can affect the algorithm behaviour. Rodriguez et al. (Rodriguez, 2012) identified 312 ISI journal articles related to the hybridization of evolutionary algorithms and Simulated Annealing algorithm, which in comparison to 123 articles used evolutionary algorithms and other metaheuristic ones, demonstrates the applicability of this system.

ANALYZING LITERATURE REVIEW

In this paper, 140 reviewed journal articles were selected based on their relevance to design and architecture and the implementation of the applied system with case studies. To create a more coherent comparison between articles in terms of methodology, four objectives of “form finding”, “Spatial Programming”, “performance-based optimization”, and “multi-objective problems” has been considered. A selection of widely used systems has been classified under these four categories which is as follows:

Form Finding and Simulation

Evolutionary modelling is a type of generative design process inspired by biological evolution to generate design solutions. A key factor in this strategy is how genes are used to provide design solutions. In literature review analysis areas of Parametric Modelling, agent-based systems (geometry optimization, and topology finding), shape grammar, graph grammar, and cellular automata were identified.

1- Parametric Modelling: The main feature of Parametric Modelling (PA) is the possibility of re-building and modification based on varied parameters. The implementation of this process in design computer programming such as Grasshopper and Dynamo is approximately similar. Parametric modelling was first invented by Rhino, a computer-aided design software developed by Robert McNeel and Associates. The key advantage of parametric modelling is, when setting up a 3D geometric model, the shape of model geometry can be changed as soon as the parameters such as the dimensions or curvatures are modified; therefore there is no need to redraw the model whenever it needs a change (Feng Fu, 2018). Parametric design allows designers to focus on formative and generative design using ‘advanced parametric applications viz., Grasshopper, CATIA, and Generative Components through scripting (Williams, 2014).

2- Agent-based systems: Agent Based Modelling and Simulation (ABMS) refers to a category of computational models invoking the dynamic actions, reactions and intercommunication protocols among the agents in a shared environment, in order to evaluate their design and performance and derive insights on their emerging behaviour and properties (Abar, 2017). When optimizing geometric patterns with evolutionary algorithms such as GA, rather simplified variables are used, since complex shapes create multiple parameters, effecting the process of finding the optimal solution. Agent-based systems can address this issue by allowing the morphing of geometry with a few agent points (Yi Y. K., 2015).

3- Shape Grammar: Knight and Stiny (Knight, 2015), extend the application of Shape Grammar in both design process and Making. “Making is Doing and Sensing with Stuff to make Things”. They modify algebras for the materials (basic elements) of shapes to define algebras for the materials of objects, or things. Figure 2 illustrates the knotting grammar inspired from *khipu* , the knotted strings made by the Incas as a physical recordkeeping and communication language. And the repetition rule model based on this idea. “*The idea is to capture the salient properties of stuff and things in actual making, so that manipulating stuff and things can be described as computation*”.

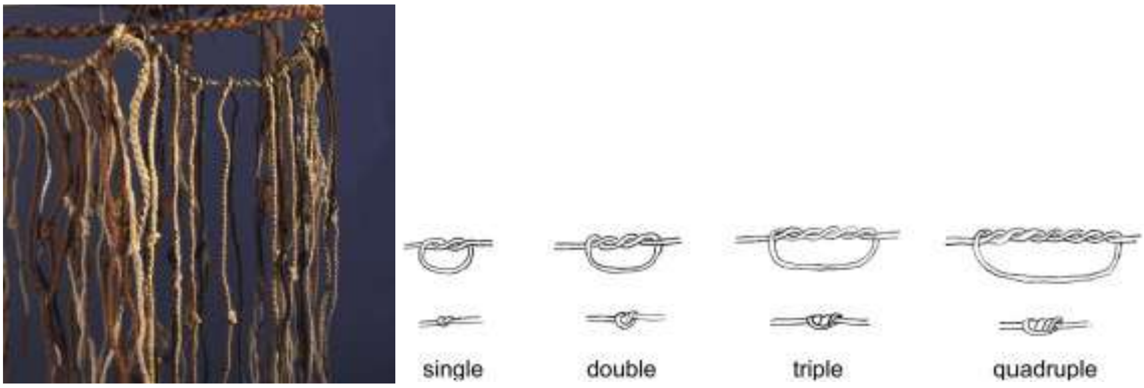

Figure 2: Khipu, and Single and multiple overhand knots model, Source (Knight, 2015)

4- Graph Grammar: Lee et. Al (Lee J. H., 2017), apply A graph grammar consists of links and nodes that are used to analyse the structural and functional relations required for generating designs. Based on authors' opinion, the two best-known approaches to computational analysis in architecture are concerned with the 'syntax' of space and the 'grammar' of form. The combined process relies on three connected processes as can be seen in Table 3.

Table 3: Node, link and shape, three process of Graph Grammar, Source (Lee J. H., 2017)

	Node	Link	Shape
Theory	Space Syntax	Space Syntax, Shape Grammar	Shape Grammar
Configuration	Functional space (vocabulary)	Functional relationship (grammar)	Formal properties (sentence/structure)
Example			

In the mentioned article, authors applied Graph Grammar to Wright's architecture, and analysed nineteen Prairie houses, typical of the Wright's work. Also, they used massing grammar to illustrate the overall form of the design. Figure 3 depicts four generation of Prairie house after the final configuration.

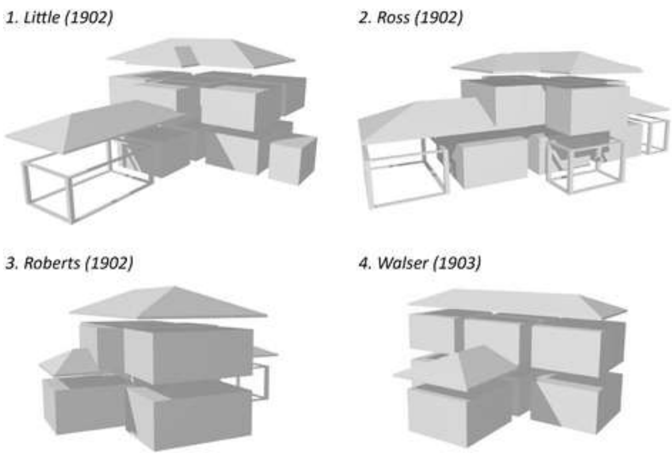

Figure 3: Massing of four Prairie houses, Source (Lee J. H., 2017)

5- Cellular Automata: this approach has been applied in generating high density residential buildings by SalmanKhalili Araghi and Rudi Stouffs (Araghi, 2015). The major characteristic of a CA generative system is to produce a vast number of solutions and generate complex morphologies by applying simple rules to cope with the majority of constraints. In this paper, CA is applied to address three element of density, accessibility and natural light in the architectural context. According to authors, "*the majority of CA applications in architecture perform conceptual form generation, allowing designers to explore a variety of results from which they can select potential solutions*". Also, cellular automata and shape grammars have the potential to be employed in a complementary way in the early stage of the self-generating design process (Speller, 2007). Araghi and Stouffs, implemented their system in three-dimensional modelling software Rhinoceros®, and programmed the CA rules in RhinoScript. Figure 4 illustrates three generated residential blocks, addressing density, accessibility and natural light.

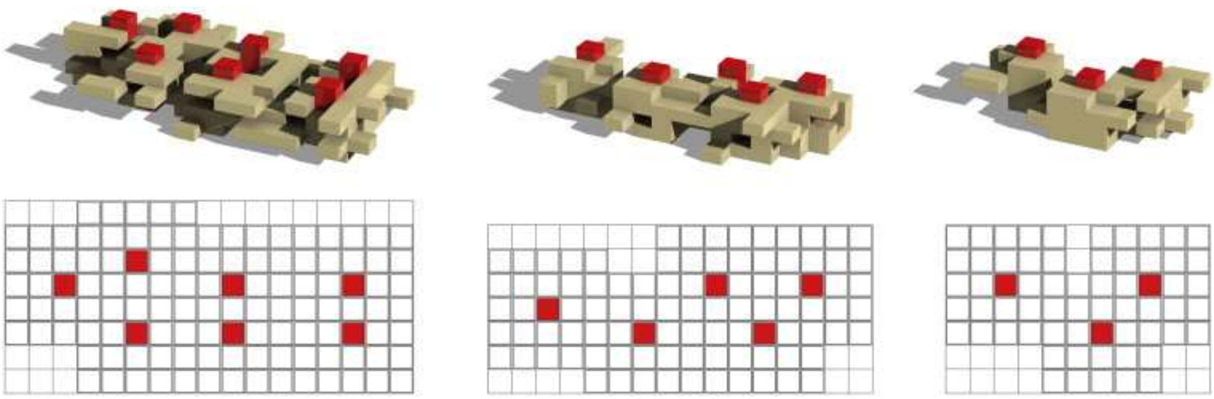

Figure 4: Three samples of generated blocks and their footprints. Source (Araghi, 2015)

Layout Generation

The problem of spatial configuration is concerned with finding suitable locations for a set of interrelated objects that meet design requirements and maximize design quality according to design preferences (Chatzikonstantinou, 2014). Spatial programming (SP) is a research field in which the process of arranging spatial elements and issues such as distance, proximity, or other functions are important. According to Gero (Gero, 1997), an SP problem is NP-complete[1] and presents all the difficulties associated with this class of problems, thus could be solved efficiently by a non-deterministic algorithm. The use of evolutionary algorithms in space planning problems has been explored since the early 90´s. Based on Calixto and Celani research (Calixto, 2015), the evolutionary methods of genetic algorithm, genetic programing, evolutionary strategies, interactive evolutionary algorithm and parallel genetic algorithm have been applied to generate plans or evaluate it, and has been combined with other methods such as shape grammar, graph theory, and adjacency matrix.

Layout design optimization can be classified into topological and geometric constraint based on constraint type. Topological constraints are defined as a hierarchical relationship of spatial elements such as proximity, non-proximity, and proximity between spaces. Geometric constraints are defined by plane, length, width, or spatial direction. Guo and Lee (Guo, 2016) applied multi-agent topology finding system and an evolutionary optimization process to address the issue of spatial layout modelling and the multi-floor topology. The Multi-agent system represent rooms as points without having volumes and shapes. Therefore, to optimize the generated model, this paper used a grid system for the conversion. Figure 5 illustrates the proposed process by the authors.

[1] A problem is called NP (nondeterministic polynomial) if its solution can be guessed and verified in polynomial time; nondeterministic means that no particular rule is followed to make the guess.

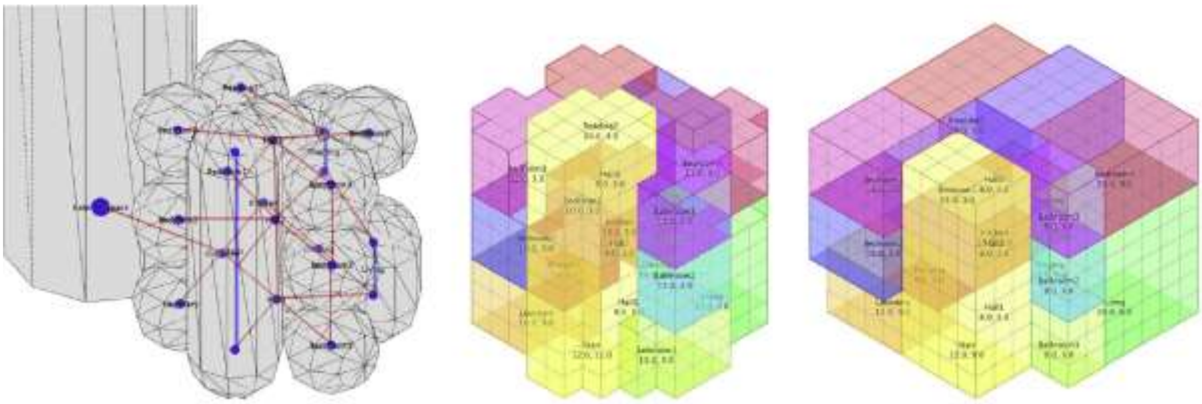

Figure 5: Left: generated multi-agent layout. Middle: converted grid system. Right: optimized grid system, source (Guo, 2016)

One of the important issues that should be considered in the application of evolutionary methods is the proper use of related tools such as programming languages or related software and plugins. Reinhard Koenig (Koenig, 2015), developed open source library for computational planning synthesis, called CPlan which enables optimization of synthesized spatial configuration. The aim of the library is to provide an easy to use programming framework for people with basic programming knowledge. This open source has basic geometry objects with a computational geometry library as well as a geometry viewer with corresponding mouse interaction. Different sections of this source are: 1- geometry library, viewer and mouse interaction, 2- computational analysis, graph measure to calculate centrality measures for street networks, Isovists field calculations, view field properties, visual centralities, and solar analysis, 3- generative methods, 4- synthesis methods, and 5- Visualization. Figure 6 depicts Cplan software prototype.

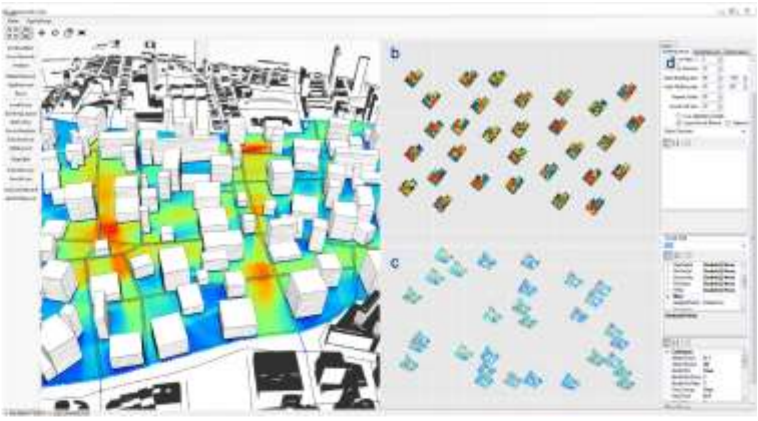

Figure 6: Software prototype showing the main areas of the synthesis systems user interface, source (Koenig, 2015)

Performance-based optimization

The significant growth in applying optimization methods to improve building performance has provided a wide range of research studies, the development of evolutionary approaches, and appropriate tools in this field. “Performance-driven architectural design” emphasizes on integrated and comprehensive optimization of various quantifiable performances of buildings. This approach takes a holistic view towards ecological and environmental performances of buildings while ensuring that the functions and aesthetics of the design are not overlooked (Xing Shi, 2013). However, most studies have concentrated on the performance aspect or techniques of computational optimization. According to Huang et al. (M.F. Huang, 2015) Performance-based design optimization (PBDO) is the combination of state-of-the-art performance-based engineering and a computational design optimization technique into an automated and synthesized design platform that aims to minimize the structural or life-cycle cost for buildings subject to natural hazards such as severe earthquakes and extreme windstorms. Searching PBDO in literature review, most studies have applied this approach to optimize life cycle cost, structural efficiency, and material cost, published in the field of engineering. In architecture domain, Shi and Yang (Xing Shi, 2013) emphasize on developing an effective technique to conduct performance-driven design and optimization from the perspective of architects. Conventionally, performance-driven optimization processes, regarding energy or structural efficiency, is implemented after the conceptual design phase by design programming tools' experts, despite numerous unified tools for simultaneous designing and BPS implementation. Architects and designers still prefer design tools such as ArchiCad, Sketchup, Revit, Rhino, and Maya, as they support the concept of a sketch and the freedoms associated with design tools (Negendahl, 2015). To integrate architects' preferences in Performance-driven architectural design, Shi and Yang selected Rhinoceros and Grasshopper (graphical algorithm editor) as a suitable platform for architects and established three performance simulation programs, namely Ecotect, Radiance, and EnergyPlus, in Rhino. Simulation results can be automatically fed back to the modelling program to guide the design optimization controlled by certain algorithms. Thus, the key to the workflow is a data exchange and communication system to control the entire design and analysis process.

Due to the significant importance of sustainability issues, designing nearly zero-energy or energy efficient buildings with Building Performance Simulations (BPSs) approaches has gained considerable attention. Table 4 classifies some of the selected articles related to performance-based optimization based on their methods, tools, and conclusion.

Table 4: Summary of selected articles regarding performance-driven optimization

REFERENCE	APPROACH	METHOD	CONCLUSION
(MIN-YUAN CHENG, 2014)	Application of evolutionary algorithm (EMARS), and artificial intelligence (AI) model, to efficiently predict the for assessing buildings energy performance	Examine the EMARS on 12 building forms simulated in Ecotect simulation software, evaluating relative compactness, surface area, wall area, roof area, overall height, building orientation, glazing area, and glazing distribution.	Surface and roof area are the most important impactful factors in heating load (HL). Cooling load (CL) in controlled by 6 out of 7 factors. CL and HL have a weak correlation with the compactness factor.
(NEGENDAHL, 2015)	building performance simulations (BPSs) in the early design stage	The assessment of user integration, and model integration (concerning computational automation processes)	integrated dynamic models may combine a design tool, a visual programming language and a BPS to provide better support for the designer during the early stages of design
(MOHAMED HAMDY, 2016)	Comparison of seven commonly-used multi-objective evolutionary optimization algorithms in solving the design	Comparison of (pNSGA-II), (MOPSO), (PR_GA), (ENSES), (evMOGA), (spMODE-II), (MODA), for a case study house in Helsinki,	provide an overall view of the performance and behaviour of these algorithms based on which researchers can make a choice for their specific

	problem of a nearly zero energy building (nZEB)	Finland using a complex energy model.	problem. tests were performed on one selected building energy model. The effect of building models on the test results has not been evaluated.
(MOHAMMAD SAFFARI, 2017)	a simulation-based optimization methodology	coupling EnergyPlus and GenOpt with an innovative enthalpy-temperature (h-T) function to define the optimum PCM peak melting temperature to enhance the cooling, heating, and the annual total heating and cooling energy performance of a residential building in various climate conditions based on Köppen-Geiger classification.	choosing the phase change materials (PCM) melting temperature in different climate conditions is a key factor to improve the energy performance in buildings.
(PATRICK SHIEL, 2018)	identify the groups of influential parameters within a design stage building energy performance simulation (BEPS) model and determine quantitatively, how influential these groups might be on the predicted energy usage	real world BEPS models developed for real world buildings. Use of Revit BIM and modification of data based on laser scan of the actual building. Derive output from Sketchup and OpenStudio in the EnergyPlus IDF format.	how a modeller has interpreted various aspects of a building's design has been acknowledged in the Literature affects model accuracy.
(WORTMANN, 2019)	architectural design optimization (ADO), considering objectives of structure, building energy, and daylight	Comparison of meta-heuristic, direct search, and model-based methods. Use of Opossum as the first established model-based optimization tool.	for practical, simulation-based, and time-intensive ADO problems with modest evaluation budgets—a global model-based method such as RBFOpt is the most likely to yield the best results.
(ABHISHEK SANJAY JAIN, 2020)	Evaluates sustainability options simulated and analysed together including Phase change materials (PCM), green roof and a cool roof. Use of GA for optimization. Parametric analysis.	Central Library of IIT Delhi is modelled for simulation using DesignBuilder, then energy efficiency analysed in EnergyPlus. Optimization with genetic algorithm.	The energy performance of an existing infrastructure may be highly improved by optimization of its inherent design parameters.

Multi-objective design Problems

Architectural design is a multifaceted process encompassing multiple and complex qualitative (intangible) and quantitative (tangible) aspects, resulting antagonistic parameters and various constraints. Generative Evolutionary Design approaches can facilitate the process of achieving optimal solutions by exploring a range of desirable alternatives and effectively computing time-consuming tasks. This approach and the related design programming tools have been successfully applied in recent studies. However, there is still extensive criticism over design automation systems due to inconsistencies of fully-automated evaluation process. To address these challenges, different approaches have been proposed, including the use of "Hybrid evolutionary algorithms" and "Interactive-generative design systems", with the key aim of allowing the designer to consider tangible and intangible aspects in one environment in the early design stage. In other words, an integrative generative evolutionary framework to generate design alternatives, evaluate their desirability, and perform optimization (to either obtain the optimum solution or gain insight into solution space) which architects can implement without the help of computer programmers, can answer the contemporary design processes. Two recent research studies in performance-based conceptual building designs with different approaches, one with the application of fully-automated design process, through co-simulation, and the other one with adopting interactive optimization technique, is presented.

1- Yi (Yi H. , 2020) mentions the pivotal role of incorporating computerized performance simulation into the design process in bridging the gap between design and engineering. In this paper, Co-simulation approach as a holistic view via the modular composition of different simulators or the hybridization of algorithms is described. Also, it proposes a computational framework that can visualize and evaluate space occupancy, energy use, and generative envelope design given a space outline. Visual programming language (VPL) of Grasshopper (GH) for Rhino is used for full integration and automation of the design process. Also, for optimization criteria built-in GH components and Phyton (IronPython) scripts was applied. The co-simulation, the Building Controls Virtual Test Bed, Energy Plus, and Radiance were interfaced in Rhino, and the agent-based model (ABM) approach and Gaussian process (GP) were applied to represent random human behaviour. Figure 7 illustrates the Scheme of Agent-Based Model-based PBD automation process.

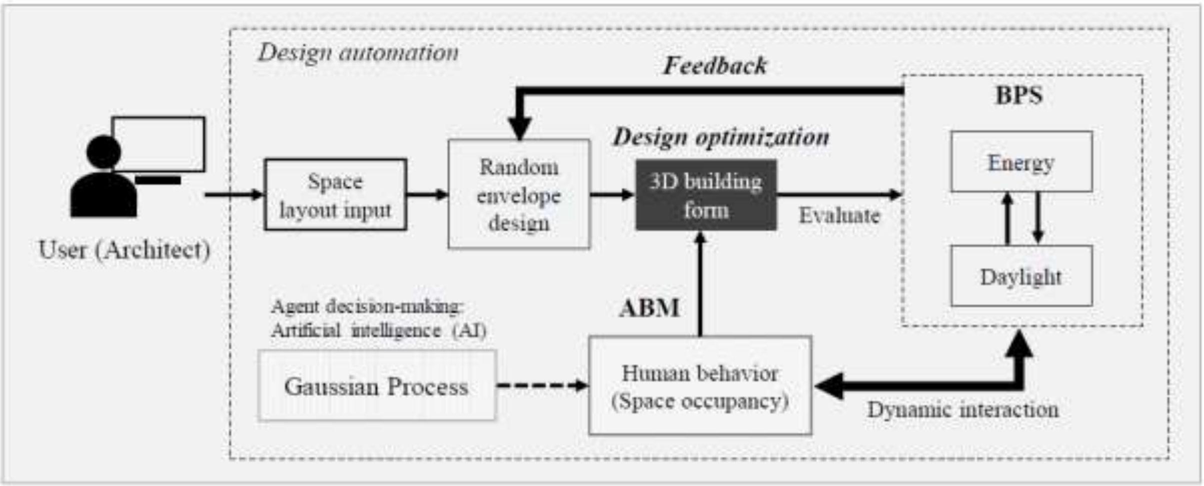

Figure 7: Scheme of ABM-based PBD automation process, source: (Yi H. , 2020)

2- Nathan and Brown (Nathan C. Brown, 2020) mention the necessity of conducting more research in analyzing the effect of interactive optimization techniques on design process. To analyse the feedback, this study engages 34 experts in various design fields to generate a roof structure for an athletic centre which has been searched among the restored parametric models. The design objectives include minimizing energy use intensity, total energy, total structural weight, and total structural weight per area. The results of their survey in four different environments, free, feedback, interactive optimization, and automated optimization, demonstrates that while designers found the interactive approach more effective, due to the lack of adequate knowledge, they had many difficulties applying the method. Also, the available estimated performance data would lead to more efficient solutions. Figure 8 illustrates the classification of four different analysed settings based on their potentials in a range from producing more performance-based solutions to allowing more diversity and creativity.

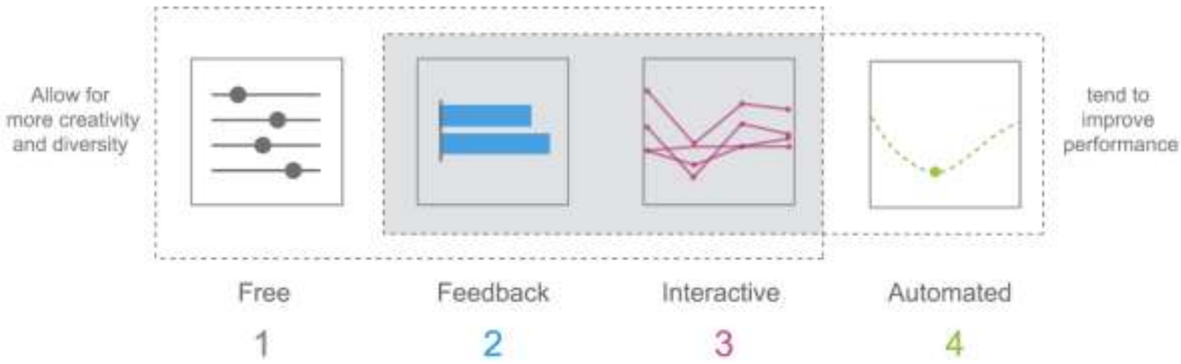

Figure 8: results of the study analyzing the effect of interactive optimization process based on designers' feedback. Source (Nathan C. Brown, 2020)

DISCUSSION and CONCLUSION

The analysis of 140 studied journal articles in four categories of form finding, Spatial Programming, Performance-based optimization, Multi-objective optimization can be classified in the following topics:

1- Applicability of meta-heuristic evolutionary algorithms in the conceptual phase in terms of performance, creativity, and diversity. There is a plethora of studies applying various algorithms, such as various types of GA (VEGA, MOGA, and so on), simulated annealing, particle swarm optimization, and so on, in design problems. However, most studies in this category lack a holistic outlook towards involved various design problems, examining the evolutionary algorithm mainly in performance-based optimization processes with few variables. There is a diverging viewpoint regarding the efficacy of GA in solving multi-objective design problems.

2- Analyzing creativity and diversity in generating design solutions based on precedent alternatives. Although many studies have applied generative approaches such as Shape grammar, Cellular automata, and Genetic Algorithm to produce design solutions coherent to a designated style, the degree of creativity and diversity in the generated solutions has less been analysed. In the selected studies used agent-based model approach, a consensus exists over its effectiveness to handle multi-objective design problems. However, the number of research in this topic is less than other evolutionary approaches.

3- Shape Optimization: topology and shape optimization are mainly studied in performance-based optimization with emphasize on iso-geometric, aerodynamic, structural and energy efficient shape optimization with rather simplified geometric relations. Fewer studies in the field of architecture address the qualitative properties of optimal topologies.

4- Conceptual design tools with Co-simulation possibility. Although conceptual design tools have been used extensively during the past twenty years, user-friendly interfaces that can offer flexible solutions for grouped design problems and enables simultaneous evaluation processes are still in their early development phase.

5- Multi-objective interactive approaches in the conceptual design phase: Interactive Generative Evolutionary systems that support designers in the early design phase has received little consideration in the literature review.

6- Integrative framework in a parametric design program: the need for programming/ scripting Performance criteria, environmental constraints, and design criteria in a visualized parametric design simulation framework is apparent.

7- Hybridization: the need to study effective methods of hybridization through combing evolutionary algorithms or various evolutionary approaches. Each of these methods should be compared in terms of outcome quality, performance, creativity, and diversity.

In this research, a descriptive-analytical approach was applied to assess literature review in generative, evolutionary, and hybrid approaches in the field of architecture. A seven-step model was proposed to thoroughly select related literature, analyse them, and classify the results based on the evaluation. The assessment demonstrates a growing interest in Hybrid approaches through the last six years from 2014 to 2020. However, most of these studies aim performance-based optimization with less or no regard to creativity criteria in design. The highest numbers of studies in form finding area applied shape grammar as an analytical generative method. Also, during the studied period, the number of studies that combine shape grammar with other evolutionary algorithms has increased, which shows the growing attention towards integrative comprehensive systems. Figure 9 depicts the comparison between 2014 and 2015 articles in terms of the applied methods. The highest number of studies were related to Hybrid systems and genetic algorithm.

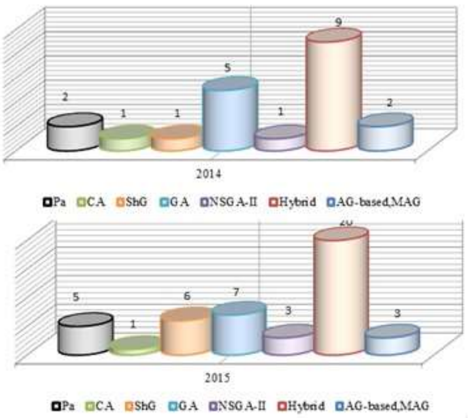

Figure 9: comparison of articles in 2014 and 2015 in terms of applied technique

Figure 10 shows the number articles based on the applied technique through the studied period of time.

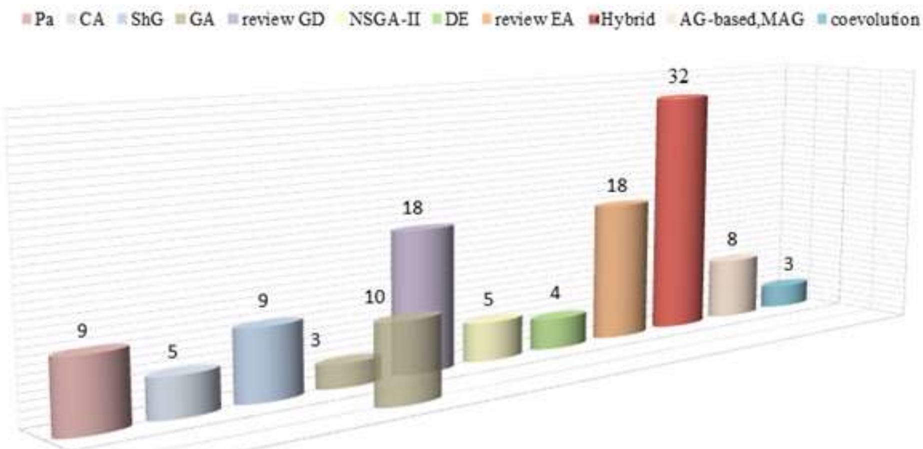

Figure 10: applied techniques in the selected literature review between 2014 and 2020.

Based on the results of analytic literature review, a research model is proposed (Figure 11) that provides a foundation for future studies in this area. Prior knowledge in design process, and various evolutionary applied approaches, as well as comprehensive definition of design objectives, problems, and constraints is necessary for achieving diverse, creative solutions.

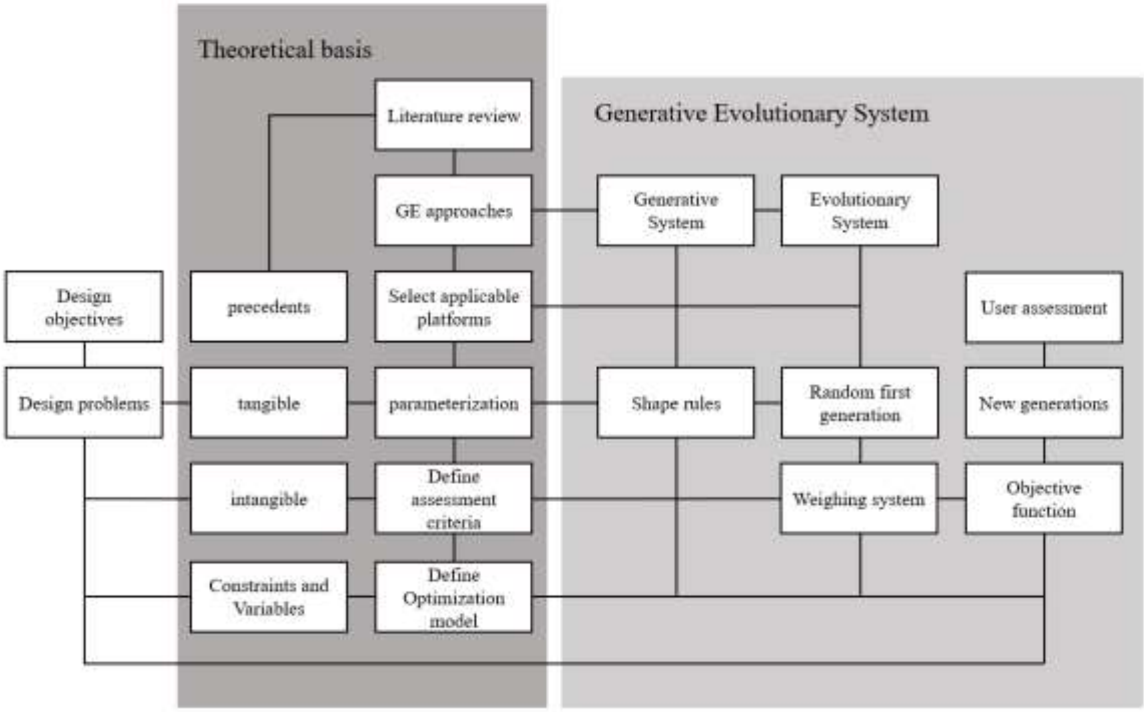

Figure 11: Study model for an integrative Generative evolutionary system

REFERENCES

Abar, S. T. (2017). Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review, 24*, 13-33. doi:https://doi.org/10.1016/j.cosrev.2017.03.001

Abhishek Sanjay Jain, P. S. (2020). Thermal energy performance of an academic building with sustainable probing and optimization with evolutionary algorithm,. *Thermal Science and Engineering Progress, 17*. doi:https://doi.org/10.1016/j.tsep.2019.100374

Araghi, S. K. (2015). Exploring cellular automata for high density residential building form generation. *Automation in Construction, 49*, 152-162. doi:https://doi.org/10.1016/j.autcon.2014.10.007

Bentley, P. (1999). Aspects of evolutionary design by computers. In Advances in Soft Computing (pp. 99-118). Springer, London.

Beyer, H. G. (2002). Evolution strategies–A comprehensive introduction. *Natural computing, 1 (1)*, 3-52. doi:https://doi.org/10.1023/A:1015059928466

Calixto, V. &. (2015). literature review for space planning optimization using an evolutionary algorithm approach: 1992-2014. *SIGRADI 2015 [Proceedings of the 19th Conference of the Iberoamerican Society of Digital Graphics*, (pp. 662-671). Florianópolis, SC, Brasil.

Ceccato, C. (1999). The architect as toolmaker: computer-based generative design tools and methods. *CAADRIA '99* (pp. 295-304). Shanghai (China): Proceedings of The Fourth Conference on Computer Aided Architectural Design Research in Asia / ISBN 7-5439-1233-3].

Chan, K.H., F. J. (2002). An Evolutionary Framework for Enhancing Design. In G. J.S., *Artificial Intelligence in Design* (pp. 383-403). Dordrecht: Springer. doi:https://doi.org/10.1007/978-94-017-0795-4_19

Chatzikonstantinou, I. (2014). A 3-dimensional architectural layout generation procedure for optimization applications: DC-RVD. *Thompson, Emine Mine (Ed.), Fusion - Proceedings of the 32nd eCAADe Conference* (pp. 287-296). Newcastle upon Tyne, England, UK: olume 1, Department of Architecture and Built Environment, Faculty of Engineering and Environment.

Eiben, A. E. (2015). *Introduction to evolutionary computing* (2nd edition ed.). Berlin, Heidelberg: springer. doi:https://doi.org/10.1007/978-3-662-44874-8

Feng Fu. (2018). *Design and Analysis of Tall and Complex Structures*. Butterworth-Heinemann. doi:https://doi.org/10.1016/B978-0-08-101018-1.00006-X

Fogel, L. J. (1963). *Biotechnology: Concepts and Applications*. Englewood Cliffs: NJ: Prentice Hall.

Frazer, J. (1995). *An Evolutionary Architecture*. Architectural Association publications, Themes VII.

Frazer, J. H. (2002). Generative and evolutionary techniques for building envelope design. *In Soddu, C (Ed.) Generative Art 2002* (pp. 3.1-3.16). Italy, Milan: 5th International Conference GA2002. Generative Design Lab.

Gero, J. K. (1997). Learning and re-using information in space layout planning problems using genetic engineering. *Artificial Intelligence in Engineering, 11*(3), 329-334. doi:https://doi.org/10.1016/S0954-1810(96)00051-9

Gong, W. C. (2008). Enhancing the performance of differential evolution using orthogonal design method. *Applied Mathematics and Computation 206(1)*, 56-69. doi:https://doi.org/10.1016/j.amc.2008.08.053

Gu, Z. T. (2006). Capturing aesthetic intention during interactive evolution. *Computer-Aided Design, 38(3)*,, 224-237. doi:https://doi.org/10.1016/j.cad.2005.10.008

Guo, Z. (2016). Evolutionary approach for spatial architecture layout design enhanced by anagent-based Topolgy finding systems. *Frontiers of Architectural Research, 6*(1), 53-62. doi:http://dx.doi.org/10.1016/j.foar.2016.11.003

Haupt, R. L. (2004). *Practical genetic algorithms*. Hoboken, New Jersey: John Wiley & Sons, Inc., .

Herr, C. M. (2016). Cellular automata in architectural design: From generic systems to specific design tools. *Automation in Construction, 72*, 39-45. doi:https://doi.org/10.1016/j.autcon.2016.07.005

HM., C. (2006). Generative Design in an Evolutionary Procedure. In T. H. Van Leeuwen J.P., *Innovations in Design & Decision Support Systems in Architecture and Urban Planning* (pp. 419-431). Dordrecht: Springer. doi:https://doi.org/10.1007/978-1-4020-5060-2_27

Ho-Huu, V. N.-T.-D.-T. (2016). An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. *Computers & Structures, 165*, 59-75. doi:https://doi.org/10.1016/j.compstruc.2015.11.014

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* Ann Arbor: The University of Michigan Press. Retrieved from Google Scholar

J., G. (1995). Interactve Evolutionary Algorithms in Design. *Artificial Neural Nets and Genetic Algorithms*, 227-230. doi:https://doi.org/10.1007/978-3-7091-7535-4_60

Janssen, P. F. (2005). Generative evolutionary design: A framework for generating and evolving three-dimensional building models. *Innovation in Architecture, Engineering and Construction,, 1*, 35-44.

Kicinger, R. a. (2005). Parameterized versus Generative Representations in Structural Design: An Empirical Comparison. *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation* (pp. 2007–2014). New York, NY, USA: Association for Computing Machinery. doi:https://doi.org/10.1145/1068009.1068344

Knight, T. &. (2015). Making grammars: from computing with shapes to computing with things. *Design Studies, 41*, 8-28. doi:https://doi.org/10.1016/j.destud.2015.08.006

Koenig, R. (2015). CPlan: An Open Source Library for Computational Analysis and Synthesis: An Open Source Library for Computational Analysis and Synthesis. *In eCAADe 2015-real time-extending the reach of computation: proceedings of the 33rd International Conference on Education and Research in Computer Aided Architectural Design in Europe. 1*, pp. 245-250. Vienna, Austria: eCAADe. doi:https://doi.org/10.3929/ethz-a-010522005

Lee, J. H. (2017). A Combined Plan Graph and Massing Grammar Approach to Frank Lloyd Wright's Prairie Architecture. *Nexus network journal, 19(2)*, 279-299. doi:https://doi.org/10.1007/s00004-017-0333-0

Lee, L. H. (2008). Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem. *European Journal of Operational Research189(2),*, 476-491. doi:https://doi.org/10.1016/j.ejor.2007.05.036

Li, W. (2004). Using genetic algorithm for network intrusion detection. *Proceedings of the United States department of energy cyber security group, 1,*, 1-8.

Limbourg, P. &. (2008). Multi-objective optimization of generalized reliability design problems using feature models—A concept for early design stages. *Reliability Engineering & System Safety, 93(6)*, 815-828. doi:https://doi.org/10.1016/j.ress.2007.03.032

M.F. Huang, Q. L. (2015). Performance-based design optimization of tall concrete framed structures subject to wind excitations. *Journal of Wind Engineering and Industrial Aerodynamics, 139*, 70-81. doi:https://doi.org/10.1016/j.jweia.2015.01.005

Min-Yuan Cheng, M.-T. C. (2014). Accurately predicting building energy performance using evolutionary multivariate adaptive regression splines. *Applied Soft Computing*, 178-188. doi:https://doi.org/10.1016/j.asoc.2014.05.015

Mohamed Hamdy, A.-T. N. (2016). A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. *Energy and Buildings, 121*, 57-71. doi:https://doi.org/10.1016/j.enbuild.2016.03.035

Mohammad Saffari, A. d. (2017). Simulation-based optimization of PCM melting temperature to improve the energy performance in buildings,. *Applied Energy, 202*, 420-434. doi:https://doi.org/10.1016/j.apenergy.2017.05.107

Mühlenbein, H. &.-V. (1993). Predictive models for the breeder genetic algorithm i. continuous parameter optimization. *Evolutionary computation, 1(1)*, 25-49. doi:https://doi.org/10.1162/evco.1993.1.1.25

Nariman-Zadeh, N. A. (2005). Inverse modelling of multi-objective thermodynamically optimized turbojet engines using GMDH-type neural networks and evolutionary algorithms. *Engineering Optimization, 37(5),*, 437-462. doi:https://doi.org/10.1080/03052150500035591

Nathan C. Brown. (2020). Design performance and designer preference in an interactive, data-driven conceptual building design scenario,. *Design Studies, 68*, 1-33. doi:https://doi.org/10.1016/j.destud.2020.01.001

Negendahl, K. (2015). Building performance simulation in the early design stage: An introduction to integrated dynamic models. *Automation in Construction*, 39-53. doi:https://doi.org/10.1016/j.autcon.2015.03.002

Nguyen, A. T. (2014). A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy, 113*, 1043-1058.

Oxman, R. (1990). Prior knowledge in design: a dynamic knowledge-based model of design and creativity. *Design Studies, 11*(1), 17-28. doi:https://doi.org/10.1016/0142-694X(90)90011-Z

Park, H. S. (2007). Evolutionary design of hybrid self-organizing fuzzy polynomial neural networks with the aid of information granulation. *Expert systems with Applications33(4),*, 830-846. doi:https://doi.org/10.1016/j.eswa.2006.07.006

Patrick Shiel, S. T. (2018). Parametric analysis of design stage building energy performance simulation models,. *Energy and Buildings, 172*, 78-93. doi:https://doi.org/10.1016/j.enbuild.2018.04.045

Poon J., M. M. (1996). Emergent Behaviour in Co-Evolutionary Design. In J. S. Sudweeks, *Artificial Intelligence in Design '96* (pp. 703-722). Dordrecht: Springer. doi:https://doi.org/10.1007/978-94-009-0279-4_37

Rechenberg, I. (1978). Evolutionsstrategien. In R. U. Schneider B., *Simulationsmethoden in der Medizin und Biologie. Medizinische Informatik und Statistik, vol 8* (pp. 83-114). Berlin, Heidelberg: Springer. doi:https://doi.org/10.1007/978-3-642-81283-5_8

Rian, I. M. (2014). Tree-inspired dendriforms and fractal-like branching structures in architecture: A brief historical overview. *Frontiers of Architectural Research, 3(3)*, 298-323. doi:https://doi.org/10.1016/j.foar.2014.03.006

Rodriguez, F. J.-M. (2012). Hybrid metaheuristics based on evolutionary algorithms and simulated annealing: taxonomy, comparison, and synergy test. *EEE Transactions on Evolutionary Computation, 16(6),* , 787-800.

Rosenman, M. A. (1999). Evolving designs by generating useful complex gene structures. In P. J. Bentley, *Evolutionary Design by Computers* (pp. 345-364). San Francisco, California : Morgan Kaufmann.

Sims, K. (1994). Evolving virtual creatures. *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (pp. 15-22). New York, NY, USA: Association for Computing Machinery. doi:https://doi.org/10.1145/192161.192167

Speller, T. H. (2007). Using shape grammar to derive cellular automata rule patterns. *COMPLEX SYSTEMS-CHAMPAIGN-, 17(1/2)*, 79-102.

Stiny, G. &. (1978). The Palladian Grammar. *Environment and Planning B: Planning and Design, 5*(1), 5–18. doi:https://doi.org/10.1068/b050005

Storn, R. P. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization 11*, 341–359. doi:https://doi.org/10.1023/A:1008202821328

Stouffs, R. &. (2015). Generative and evolutionary design exploration. *AI EDAM, 29(4)*, 329-331. doi:doi:10.1017/S0890060415000360

Strobbe, T. P. (2015). Toward a visual approach in the exploration of shape grammars. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AI EDAM, 29(4)*, 503-521. doi:10.1017/S0890060415000475

Thangaraj, R. P. (2011). Particle swarm optimization: hybridization perspectives and experimental illustrations. *Applied Mathematics and Computation, 217(12)*, 5208-5226. doi:https://doi.org/10.1016/j.amc.2010.12.053

Wang, L. L. (2016). Gas-liquid two-phase flow measurement using Coriolis flowmeters incorporating artificial neural network, support vector machine, and genetic programming algorithms. *IEEE Transactions on Instrumentation and Measurement, 66(5),* , 852-868.

Williams, J. L. (2014). Parametric Design Strategies for the Generation of Creative Designs. *International Journal of Architectural Computing*, 263-282. doi:https://doi.org/10.1260/1478-0771.12.3.263

Wortmann, T. (2019). Genetic evolution vs. function approximation: Benchmarking algorithms for architectural design optimization. *ournal of Computational Design and Engineering, 6*(3), 414-428. doi:https://doi.org/10.1016/j.jcde.2018.09.001

Xing Shi, W. Y. (2013). Performance-driven architectural design and optimization technique from a perspective of architects,. *Automation in Construction, 32*, 125-135. doi:https://doi.org/10.1016/j.autcon.2013.01.015

Yi, H. (2020). Visualized Co-Simulation of Adaptive Human Behaviour and Dynamic Building Performance: An Agent-Based Model (ABM) and Artificial Intelligence (AI) Approach for Smart Architectural Design. *Sustainability, 12*(16). doi:https://doi.org/10.3390/su12166672

Yi, Y. K. (2015). Agent-based geometry optimization with Genetic Algorithm (GA) for tall apartment's solar right. *Solar Energy, 113*, 236-250. doi:https://doi.org/10.1016/j.solener.2014.11.007