Advances in Computational Design, *Vol. 7, No. 2 (2022) 153-171*
https://doi.org/10.12989/acd.2022.7.2.153

A hybrid approach of generative design methods for designing tall-buildings form

Pouria Tofighi, Ahmad Ekhlassi and Morteza Rahbar*

School of Architecture and Environmental Design, Iran University of Science and Technology, Tehran, Iran

(Received August 26, 2021, Revised January 1, 2022, Accepted January 20, 2022)

Abstract. The present study aimed to find a way to create forms that can simultaneously meet several architectural requirements by applying generative design methods specifically focused on cellular automata. In other words, it is tried to find various forms of architecture that all have common features. Because of the useful features of cellular automata, we decided to use it to generate various forms, but make a relation between the discrete nature of cellular automata and the continuous nature of architecture, was the major problem of our project. To achieve this goal, three consecutive stages were designed. In the first stage, independent variables including the location of the building, the height of the building, and the building area were considered as the inputs of the model. In the second stage, after locating the building, the building's main shell was designed as a hidden geometry for the cellular automata and then the cellular automata were determined based on this shell. The main result of this research is establishing a logical relationship between the discrete geometry of the cellular automata and the continuous search space such that it creates various optimized forms. Although we specify the site plan of this project at Iran-Tehran, this research can be generalized to various design sites as well as different projects, allowing the architects to alter the cell dimensions, cell density, etc., based on their opinion and project needs.

Keywords: cellular automata; generative design; genetic algorithm; parametric design; rule-based design

1. Introduction

The design of a form in architecture has always been among the most challenging issues encountered by architects in the design process. Based on the Francis D.K. Ching's opinion in "Architecture Form, Space, & order" book, Form is a pervasive expression that has several meanings. Sometimes it denotes external appearances which can be recognized, like a table or a sofa. It can also be considered based on something's condition. For example, the forms that water has in ice or steam condition. In art and design, we use it to define the formal structure of a work. In other words, and as it is mentioned in the book, "arranging and coordinating elements and a part of a composition." Form designing is influenced by various factors such as climatic, structural, and aesthetic factors and the architect strives to find a reasonable answer to most of these needs by creating a suitable form. Studying the factors affecting the design has led to a clear and constant thought in the design process and especially in the architect's thinking. This thought process

*Corresponding author, Assistant Professor, E-mail: rahbarm@iust.ac.ir

http://www.techno-press.org/?journal=acd&subpage=7 ISSN: 2383-8477 (Print), 2466-0523 (Online)

causes the architect to neglect a large part of the problem-solving space and thus produce mostly identical forms (Benjamin *et al*. 2019). These identical forms in architecture may cause monotony in space and the skyline of cities, leading to boredom in the inhabitants of cities.

With the introduction of computers in the world of architecture, the desire to solve problems algorithmically has increased, and, consequently, this field has attracted the attention of architects. Rule-based designs are one of the new topics in architecture in the last decades such that several studies have been done in this field. Although some of these researches have good results, in some cases there are still some undiscovered realms (Rahbar 2018, Rahbar *et al*. 2022). Among these rule-based methods, we can cite generative design methods such as shape grammar (SG) and cellular automata (CA). In this paper, generative design is used to find form and express a method for how to combine cellular automata (CA) with other methods of generative design for producing a wide range of architectural forms. Such forms can both respond to the need for diversity in architecture and have the necessary features to answer architectural problems.

So far, cellular automata have been used in architectural design in various ways and many researchers have tried to find a way to use cellular automata in architecture. Despite all advantages of cellular automata in the generation of various forms, the main problem of using it in architecture is making a logical relation between the discrete nature of cellular automata and the continuous nature of architecture. In this regard, the present study aims to introduce a new method for using cellular automata in architecture by creating a volumetric shell and introducing it as the hidden geometry of a cellular automata façade system.

2. Background

2.1 Cellular automata application

In the 1940s, following the invention of the cellular automata system by John von Neumann and Stanislav Ulam, many researchers began to study or use this system in their research projects. Wolfram (2002) examined this system and its rules and discovered simple rules to produce complex shapes. John Conway created an approach called the ‘game of life’ by discovering the rules in two-dimensional space to create a specific shape process without the need for a user (Shiffman 2012).

The application and research of cellular automata did not end up with producing complex shapes and thus has been gradually used in many sciences such as structural design (Slotta *et al*. 2001, Cort’es *et al*. 2005, Funs and Pablo Jose 2001), mechanical engineering (Hejela and Kim 2001), physics (Makarenko 2019), traffic simulation (Zeng *et al*. 2021), chemistry (Natalia Menshutina *et al*. 2020), biology, and geography (Wolfram 2002). For example, a group of researchers studied the behaviour of various gases and fluids by cellular automata (Gunawan *et al*. 2017). In physics, cellular automata have been used in quantum physics (Makarenko 2019), while it has been used in other ways in biology and other sciences.

A group of researchers combined cellular automata with other methods of generative design. For example, (Cerruti *et al*. 2020) integrated cellular automata and genetic algorithms to simulate the stone-paper-scissors game, the game of life, and prisoners’ dilemmas to achieve a faster response. In this innovative method, they combined the genetic laws and the cellular automata rules governing each game. So, by creating specific goal functions and defining new neighbourhoods for each cell, they could achieve the desired answers in less time.

In architecture, like other sciences, cellular automata have been used in various fields. For the first time, researchers such as Krawczyk (2002), Herr and Ford (2015), Herr and Kvan (2007), and Coates *et al*. (1996) investigated the feasibility of using cellular automata in architecture. Later, this system was used in various fields such as form design (Araghi and Stouffs 2015, Dounas *et al*. 2017, Herr and Kvan 2005, Devetakovic *et al*. 2009, Cruz *et al*. 2016), space design (Cruz *et al*. 2017, Lee and Kim 2016, Dincer 2014), structural design (Kicinger *et al*. 2004), facade design (Caetano *et al*. 2020, Fathy *et al*. 2015, Zawidzki 2010), building evacuation simulation (Pelechano and Malkawi 2008), and pedestrian movement (Dijkstra *et al*. 2001).

2.2 Cellular automata application in architecture

As mentioned, Krawczyk and Herr were the first ones who investigated the possibility of using cellular automata in the architectural process. Krawczyk (2002) stated that the main problem of using cellular automata in architecture is the lack of vertical and horizontal continuity of cells and thus offered various solutions to solve these problems. This researcher reported that the problem of horizontal discontinuity of the cells can be overcome by changing the dimensions of the cell, adjusting the geometry of the cell, the position of the cells, and overlapping the cells. He suggested using vertical columns at the corners or centre of each cell to solve the problem of vertical cell communication. Herr and Ford (2015) also confirmed the problems proposed by Krawczyk and proposed separate solutions to these problems. This project was aimed to design a hotel by cellular automata, for which they first used the common rules governing cellular automata. Herr and Ford state that although the forms obtained from these rules are beautiful, none of them can be converted into architectural volumes. Accordingly, after this stage, these researchers tried to invent rules appropriate to the needs of architecture but failed due to the complexity of the rules and the incompatibility with the nature of the cellular automata based on the simplicity of the rules. Later, like Krawczyk, these authors adjusted the dimensions of the cells and finally stated that the use of cellular automata is possible in the initial phases of any design and for the inspiration and idea of architects.

Various projects have been conducted in the design of forms. For instance, Araghi and Stouffs (2015) designed the architectural form by considering the two criteria of lighting and access to the vertical communication core. In this study, any cell that did not have access to the communication cores in its first and second neighbourhoods or did not have a suitable position in terms of lighting was removed in the next generation, and thus various architectural forms were created. Moreover, another article discussed the improvement of urbanization rates in China. The Cellular Automata, the rules of the game of life (GOF) in three dimensions, and the rules of increasing the density and diversity of the building were used to develop housing in Suzhou Industrial Park using a design algorithm. The purpose of this design was to achieve buildings that create more density compared with all design criteria (Dounas *et al*. 2017).

As mentioned, cellular automata have been used in other fields of architecture, but it still suffers from problems such as inconsistency, inability to analyse forms, and structural problems. In the present study, an attempt was made to invent a new way of form designing by cellular automata to answer both the problems arising from the need for diversity in architecture and the appropriate climatic.

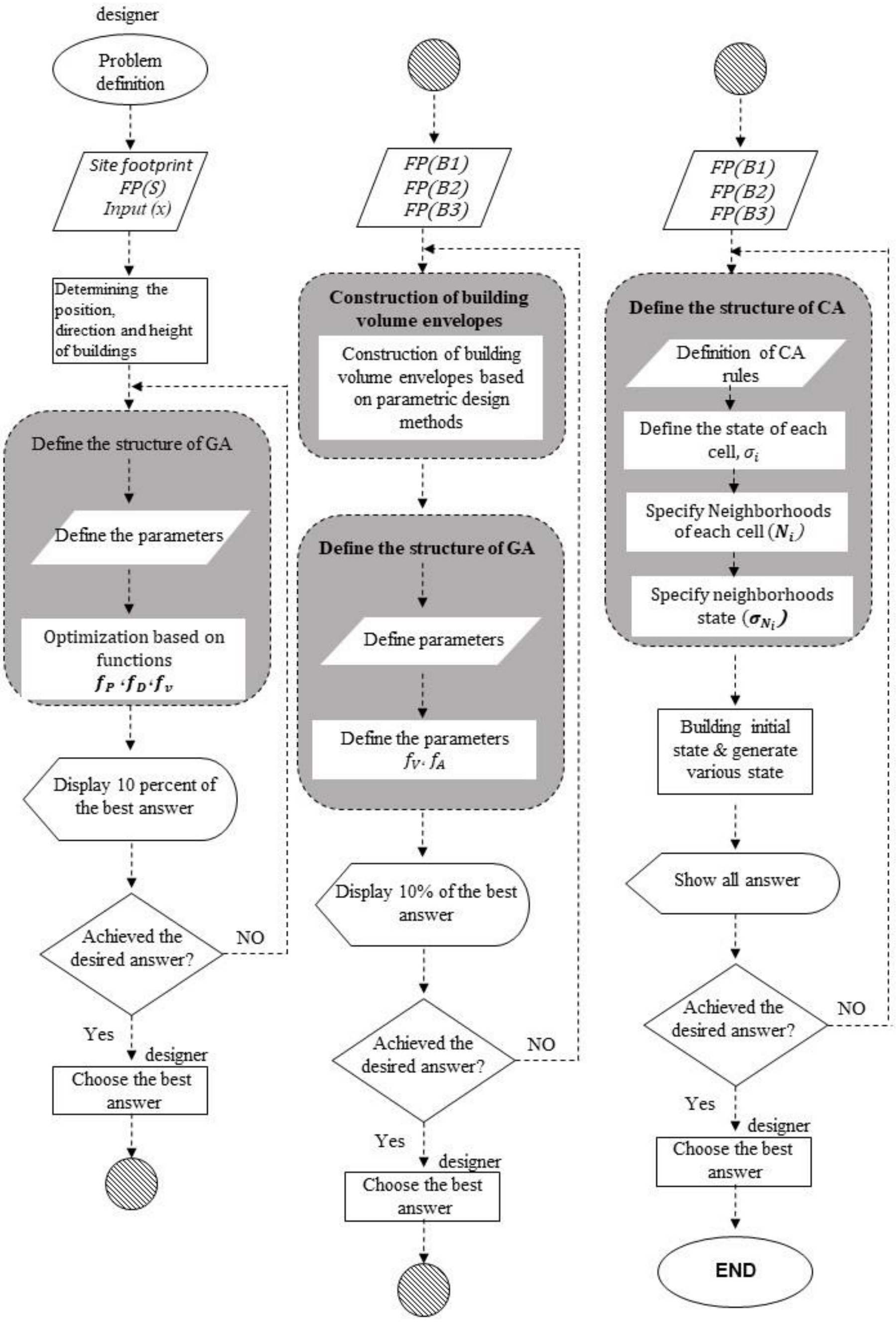


Fig. 1 Flowchart of the design algorithm

3. Method

The present study aimed to produce various tall building forms using generative design method (cellular automata) so that these forms can meet other needs of the building, such as visibility,

Table 1 List of signs and abbreviations

Signs	Description	Signs	Description
FP_{B_i}	Footprint of the i[th] Building	CA	Cellular automata
Input x	Independent input parameters	σ_i	State of the i[th] cell in cellular automata
FP(s)	Site footprint	N_i	The neighborhood of the i[th] cell in cellular automata
R_{B_i}	The amount of rotation of the i[th] building	σ_{N_i}	Neighborhood state of the i[th] cell in cellular automata
L_{B_i} ‘	Location of the i[th] building	initial state	The initial state in a cellular automata system
H_{B_i}	The height of the i[th] building	R	The rule used in cellular automata
f_P	Plaza area objective function	B_C	Dimensions of each cell
f_D	Access rate objective function	S	seed
f_V	Visibility rate objective function	G_iS	The i[th] generation of cellular automata
f_A	Building area objective function	D_eC	Cell density
GD	Generative design	ST	Step
SG	Shape grammar	H_c	Height of cells

access to building from the street, and Minimum distance of buildings from each other (based on architects' strategy which shapes plaza space) at the first step and to provide the required infrastructure for the building at the second step. As it is clear, we have a problem with more than one objective function which must be optimized simultaneously. These problems are called multi-objective function problems that cannot solve in the usual methods. In this case, we use the Wallecei plugin (version 2.6), which uses the Genetic algorithm to solve multi-objective function problems, to address our problem. Since cellular automata and the stated goals are discrete and continuous, respectively, it is essential to find a way to communicate between them. The method of the present research is based on computer modelling to find a suitable algorithm both to establish a connection between the cellular automata and the stated goals and to produce various forms. Fig. 1 shows the flowchart of the research algorithm. As can be seen, the research process consists of several different parts. Regardless of the study phase, after defining the problem, in the first step, the site plan lines and independent parameters are given as input to the algorithm. All the steps of this algorithm are described in detail in the following sections. After receiving the site's footprint FP(s), the algorithm first locates the building blocks randomly and then optimizes the parameters of the building's rotation R_{B_i}, location L_{B_i}, and height H_{B_i} based on the objective functions plaza area (A space that is located between buildings) f_P, access rate f_D and visibility rate f_V. After performing the optimization process, the best answers to this stage are presented and one of them is selected. In the next step, the algorithm uses a defined process to design the volumetric shell. In this section, the algorithm makes the initial design of the shell by taking variables such as the amount of rotation of the building floors and the amount of change in the dimensions of each floor of the building. Next, it optimizes the shell based on the objective functions f_A, f_V. In the last stage and after the finalization of the volumetric shell, the final form is designed by the cellular automata. The rules governing the cellular automata were created in two ways, one by the designer and tailored to the needs of architecture and the other by using the genetic algorithm. Finally, the architect can choose a form from among the created forms according to his/her needs. Our goal in selecting the form by the architect is to involve her/him in the design process to get better results and solve possible problems.

In summary, in this research, first, a volumetric shell is designed, followed by producing various forms using cellular automata. In this way, it is possible to communicate between cellular automata as a discrete method and architectural criteria as continuous goals and creating various forms all meeting the criteria set by the designer are achieved.

Table 1 provides a list of symbols and abbreviations used in this manuscript.

3.1 Design steps using cellular automata

As mentioned earlier, it is necessary to establish a logical relationship between the Cellular Automata (CA) and the optimization objective functions regarding the difference in their nature. It was also stated that some features of cellular automata have always limited their use in architecture. So, it is seen that the continuous nature of some objectives in architecture conflicts with the discrete nature of cellular automata. In this regard, the use of this generative design method in architecture always faces many challenges. To overcome these limitations, we designed a new method by combining several different generative design methods.

Krawczyk (2002) and Mitrovic *et al*. (2009) stated that each form of cellular automata has a hidden geometry in its field that can accommodate all the cells of a cellular automata system. Based on the above definition, it was decided to use the hidden geometry of the cellular automata. This means that instead of designing a form using cellular automata and discovering its hidden geometry, first, a volumetric shell is designed as a hidden geometry and then the cellular automata are implemented based on this geometry. This ensured that the form derived from the cellular automata is a subset of the hidden geometry and can therefore inherit all the features and characteristics of this hidden geometry. To achieve the intended goal, three consecutive steps were designed. In the first step, the location of buildings, the height of buildings, and also determining the building area on the site were designed. In the second step, the shell (hidden geometry) was designed, and then a cellular automata system was implemented. In the following, each section is described in detail.

A Sony system with a Core i5 Intel processor, 8.00 Gigabyte Ram, and NVIDIA graphic card was used in the project. It is also used Rhino 6.35 software and its plugins, Grasshopper, and python for doing these 3 steps.

3.1.1 ST1: Location of the building

In the first step, after introducing the design area (site plan), the location of the buildings, the amount of rotation of the buildings, the height of the building, and the building area on the site are determined. According to the pseudo-code presented in Fig. 2, in the first step, the system starts to run by taking the footprint of the site and ends with determining the stated items.

After specifying the site’s footprint by the user, the system first grids the site and creates cell centres that can potentially be considered as centres for buildings. The height and area of the buildings are parametrically related and the algorithm can make changes within a predetermined range to achieve the desired answer. Also, the algorithm can adjust to the position and rotation rate of each building block. So, the first answer is obtained randomly from the large space of problem answers.

	pseudo-code
1	START
2	Ask the user for PT(s)
3	Ask the user for lattice cells dimensions
4	Ask the user for a range of buildings floors.
4	Divide PT(s) by lattice cells dimensions
5	Create the center of each cell.
6	Select the center of one of the cells.
7	Optimize fitness functions by Multi-Objective Gene tic algorithm.
8	Display 10 percent of best answers.
9	Ask the user to select the best solution.
10	End

Fig. 2 Pseudo-code of the first step

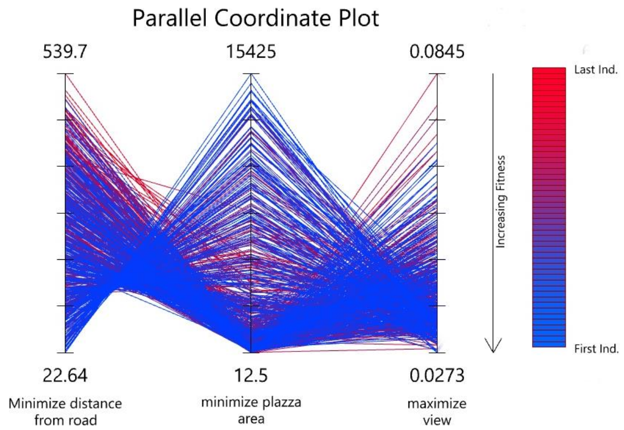


Fig. 3 All states of the first step

After this stage, the optimization operation begins to commensurate with the target functions considered by the designer and based on the genetic algorithm. In this section, three different objective functions are considered for optimization and are evaluated and optimized using the Wallacei multi-objective add-on in Grasshopper3d plugin in Rhinoceros software. These objective functions are 1) creating the best view for each building, f_V, 2) creating the closest access from each building to the communication channels of the site, f_D, and 3) creating the minimum plaza area between the buildings (according to the architect's design strategy), f_P. Modifiable variables are 1) the location of the building at the site L_{B_i}, 2) the amount of rotation of the building R_{B_i}, and 3) the height of the building H_{B_i}.

After creating 50 generations and each generation including 100 children by the Wallacei plugin multi-objective genetic algorithm, 5000 different answers were obtained (Fig. 3). About 10% of the best answers were displayed and used for the design ().

Selections' method was based on the answers' ranking depending on the average of all fitness functions. So at first, we extracted approximately 500 answers which had a rank between 1 to 500.

Table 2 Selected answer from the first step

Populations [Gen: Ind]	First objective function f_V	Second objective function f_D	Fourth objective function f_P
[00:44]	432.62	0.035	250
[16:02]	0.038	99.9	8100
[17:03]	L0.042	97.98	9375
[20:47]	0.032	148.97	4900
[22:19]	0.033	41058	12250
[25:46]	0.043	421.16	162.5
[30:9]	0.031	71.74	9662
[32:01]	0.033	31.34	13600
[42:00]	0.029	46.64	14877.5
[49:14]	0.033	101.3	7337.5

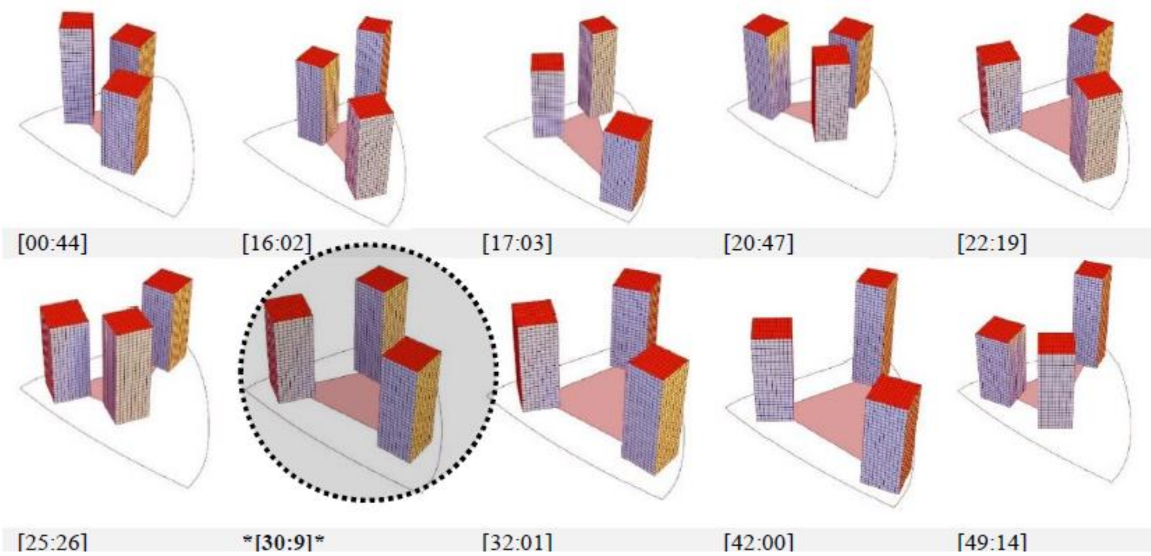


Fig. 4 Answers selected from the first step

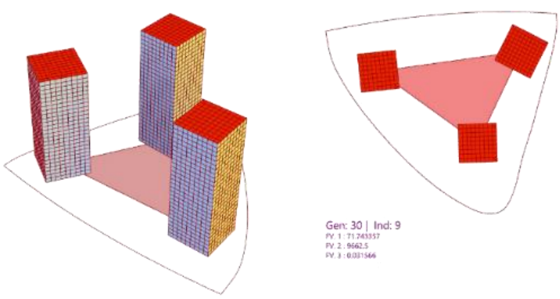


Fig. 5 The best answer selected from the first step

pseudo-code
START
Ask user for $\mathbf{FP}_{\mathbf{B_i}}$
Ask the user for a range of scale
Ask the user for a range of rotation
Create core of buildings
Create different spaces based on extending the side of the core
Create variable rotate
Create variable scale
Extrude each space based on $\mathbf{H}_{\mathbf{B_i}}$
Scale and rotate each floor based on variables dom ain.
Optimize fitness functions by Multi-objective evolu tionary algorithm
Display 10% of best solutions
Ask the user to select the preferred best solution
End
Ask user for $\mathbf{FP}_{\mathbf{B_i}}$

Fig. 6 Pseudo-code the second step

We saw that although these final solutions had acceptable fitness values, some of them could not meet our architectural purpose depending on their geometry. So based on our strategy, which we mentioned at the previous section, we let the architect enter the process and remove unacceptable solutions. The above procedure was a recursive function to achieve the best solution. Table 2 and Fig. 4 display the final ten excellent answers, and Fig. 5 demonstrates the architect's final solution to continue the design process.

3.1.2 ST2: Creating a volumetric shell

After the first stage and determining the best answer by the architect, in the second stage, the volumetric shell of the building is designed. In this section, it is tried to design a shell commensurate with the needs of the architecture and optimize it for some of the objective functions considered by the designer.

This pseudo-code is presented in Fig. 6. To perform this process, the footprint that was determined in the previous step is introduced as a new input to the algorithm. Then, the algorithm creates lines around the core of the tower. It is of note that the core of the tower occupies an average of 20% of the space on each floor. The relationship between the core area of the tower and the area of each floor is defined parametrically. In the next step, the sides resulting from the formation of the cores are extended, and by intersecting them with the perimeter lines of the building, different parts are defined. In this research, 4 different spaces are created according to the physical program. Now, after defining two objective functions (i.e., 1- the best view for each building f_v, and 2- creating the minimum infrastructure desired by the designer f_A) and two independent variables (i.e., 1- the amount of rotation of building floors and 2- the amount of floor scale), the shell of the building will be optimized. The range of rotation of the floors and its scale

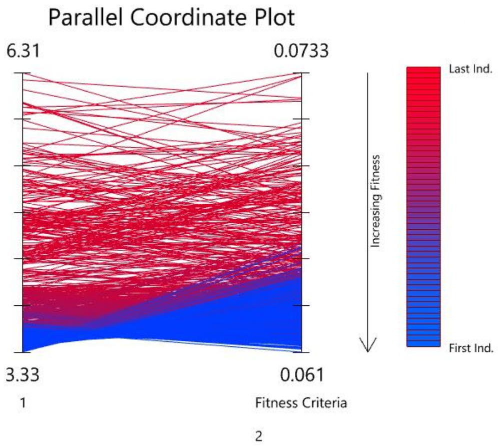


Fig. 7 All scenarios of the second step

Table 3 Specification of the selected answer of the second step

Generation [Gen; Ind]	First criterion (f_V)	Second criterion (f_A)
[32:21]	0.061	3.56
[37:21]	0.062	3.41
[39:18]	0.062	3.42
[40:01]	0.064	3.34
[41:36]	0.062	3.45
[42:01]	0.064	3.34
[44:05]	0.062	3.46
[46:43]	0.061	3.38
[49:10]	0.063	3.37
[49:35]	0.062	3.38

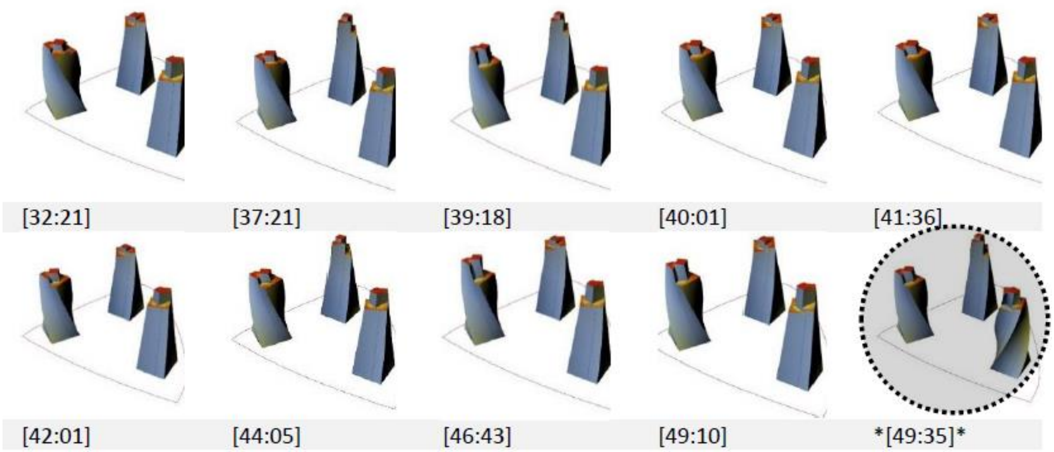


Fig. 8 Some of the best answer in the second step

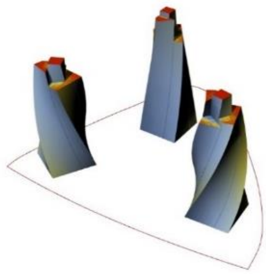


Fig. 9 The final form of the second step

is based on the designer's opinion. Similar to the first step, this step was optimized using the Wallacei plugin multi-objective evolutionary algorithm. After running the algorithm, a solution space of 5,000 different answers was obtained (Fig. 7). After prioritizing the answers according to the previous step, 10% of the best answers are displayed (based on the previous method explained) (Fig. 8), and the architect is allowed to choose one of them as a criterion for continuing the optimization process. Fig. 9 shows the option desired by the authors of the article to continue the process. Table 3 shows the characteristics of its objective functions.

3.1.3 ST3: Creating the form by cellular automata

In this step, Cellular Automata is used to create diversity forms. CA is a generative design method made by John Von Neumann to build a machine with its repeatability. Cellular automata are used to simulate biological behaviour in nature; however, as mentioned in the previous sections, this method has been used in various fields such as architecture, physics, and chemistry. Each cellular automata system in the space of [1 to n] dimensions has characteristics that cause specific features of the system. These specifications are:

• Grid: Each cellular automata system has a grid that can be one to several dimensions according to its space.

• State of each cell: In a cellular automata system, each cell has a definite state at any fixed time, which can be defined as 0 and 1 or as a range of continuous numbers. Here, the state of each cell at time t will be displayed as σ_i(t)

• Cell Neighbourhood: Each cell in the cellular automata has neighbourhoods that vary in number and type of neighbourhood according to the cellular automata space. Here, the neighbourhoods of cell i with N and the state of each neighbourhood cell at time t will be show as σ_{N_i} (t) (Neumann. J. and Burks. A. 1966).

One of the most famous systems created by cellular automata is the game created by Conway called "Game of Life" (Shiffman 2012). In this study, the Conway system, but with new customized rules are applied to design the form. The cellular automata of this research are in two-dimensional and three-dimensional spaces. So, after creating various architectural forms, the designer can choose one of them according to his/her preferences. According to the pseudo-code presented in Fig. 10, after determining the number of floors of each building and volumetric shell, the algorithm first creates a three-dimensional rectangular grid according to the dimensions specified by the architect and the number of floors of the building. Next, it removes those cells that are not in the volumetric shell. The algorithm also randomly removes some of the remaining cells according to the percentage of infrastructure specified by the designer.

Now, the first generation of cellular automata has been created, it is tried to create various forms according to the created rules. The rules of this research are defined in two ways. The first method is to create rules appropriate to the architecture and based on the concepts of GOL (Eq. (1)) by considering three issues consisting of; 1- structural issues, 2- horizontal connection, and 3- vertical connection. For example, we tried to adjust the rules following the structural needs to have no console longer than 3 meters. We also tried there are no cells without a directly horizontal or vertical connection. (Eqs. (2) and (3)). The second method is to use the evolutionary algorithm and consider creating the minimum area required for the architectural design as an objective function (Eq. (4)). According to the two methods mentioned, the following rules (Eqs. (2) and (4)) were formed in two-dimensional and three-dimensional modes to build the form.

$$\text{GOF Rules}$$

$$\begin{aligned} if\ \sigma_i(t) = 1 \rightarrow \sigma_i(\mathrm{t}+1) &= \begin{cases} 1;\ if\ 2 \leq N_i \leq 3 \\ 0; else \end{cases} \\ if\ \sigma_i(t) = 0 \rightarrow \sigma_i(\mathrm{t}+1) &= \begin{cases} 1; if\ N_i = 3 \\ 0; else \end{cases} \end{aligned} \tag{1}$$

$$\text{CA\_2D}$$

$$\begin{aligned} if\ \sigma_i(t) = 1 \rightarrow \sigma_i(\mathrm{t}+1) &= \begin{cases} 1; if\ 3 \leq N_i \leq 6\ and\ N_i(k-1) \geq 3 \\ 0; else \end{cases} \\ if\ \sigma_i(t) = 0 \rightarrow \sigma_i(\mathrm{t}+1) &= \begin{cases} 1; if\ 3 \leq N_i \leq 7\ and\ N_i(k-1) \geq 3 \\ 0; else \end{cases} \end{aligned} \tag{2}$$

$$\text{CA\_2D}$$

$$\begin{aligned} if\ \sigma_i(t) = 1 \rightarrow \sigma_i(\mathrm{t}+1) &= \begin{cases} 1; if\ 3 \leq N_i \leq 5 \\ 0; else \end{cases} \\ if\ \sigma_i(t) = 0 \rightarrow \sigma_i(\mathrm{t}+1) &= \begin{cases} 1; if\ 3 \leq N_i \leq 5 \\ 0; else \end{cases} \end{aligned} \tag{3}$$

$$\text{CA\_3D}$$

$$\begin{aligned} if\ \sigma_i(t) = 1 \rightarrow \sigma_i(\mathrm{t}+1) &= \begin{cases} 1; if\ 3 \leq N_i \leq 11 \\ 0; else \end{cases} \\ if\ \sigma_i(t) = 0 \rightarrow \sigma_i(\mathrm{t}+1) &= \begin{cases} 1; if\ 17 \leq N_i \leq 23 \\ 0; else \end{cases} \end{aligned} \tag{4}$$

4. Discussion and result

Regarding the discrete nature of cellular automata, its use in architecture has always encountered many difficulties and complexities. To deal with these issues, a combination of generative design methods (including cellular automata, parametric design, and an evolutionary algorithm) was proposed. Each cellular automaton system has characteristics such as cell size and cell density (which leads to porosity in volume) that can take different values according to the designer's opinion and needs. According to Fig. 11, the density in each system plays an important role in creating form diversity so that by increasing the considered density, the volumetric porosity and consequently its form diversity decreases. On the other hand, by decreasing this density, more

	pseudo-code
1	START
2	Get envelope shell
3	Ask the user for the dimension of CA as D & get the range of floor s
4	Create a rectangle
5	Create a 3D lattice: For _ in range(D): copy rectangle in X-axis = a For _ in range(D): Copy a in Y-axis = b For _ in range (floors): Copy b in Z-axis Create a list = Initial state For cells in the lattice: If cells in the lattice: Add cells to initial state Else: Ignore it Create variable H_c Ask the user for H_c Assign the initial states randomly and extrude them based on H_c Create a list of generations Add initial state to generations
6	Get rule of GOL Create a list for the new generation Ask the user for iteration For _ In iteration: For cells in the last list of generations: Get the neighborhoods state of it and add them together = M If cells in envelope shell and M complied with the rules : Add cells to a new generation Else: Ignore it Add new generation to generations
7	Display the last list of generations
8	End

Fig. 10 Pseudo-code of the third step

diverse forms are obtained during different generations. In this modelling, the initial state of the cellular automata plays an important role in the final form-finding so that more form variation is achieved by changing the initial state. Another noteworthy point is the same dimensions of the cell in the formation of the final form. If the dimensions of the cells are considered small, these small cells will create a form like the designed hidden shell; however, this can make it difficult for the designer to design and create an architectural plan. On the other hand, if the dimensions of these cells are considered large, the forms will bear little resemblance to the shape of the shell designed in the second step, although they can provide a good plan space for the designer.

(a) [R=2; B_C=3*3*5; S=38; G_iS=3; D_eC= 50%]

(b) [R=2; B_C=3*3*5; S=38; G_iS=3; D_eC= 70%]

(c) [R=2; B_C=3*3*5; S=38; G_iS=9; D_eC= 50%]

(d) [R=2; B_C=3*3*5; S=89; G_iS=9; D_eC= 50%]

(e) [R=4; B_C=3*3*5; S=38; G_iS= 9; D_eC= 40%]

(f) [R=4; B_C=3*3*5; S=72; G_iS= 3; D_eC= 40%]

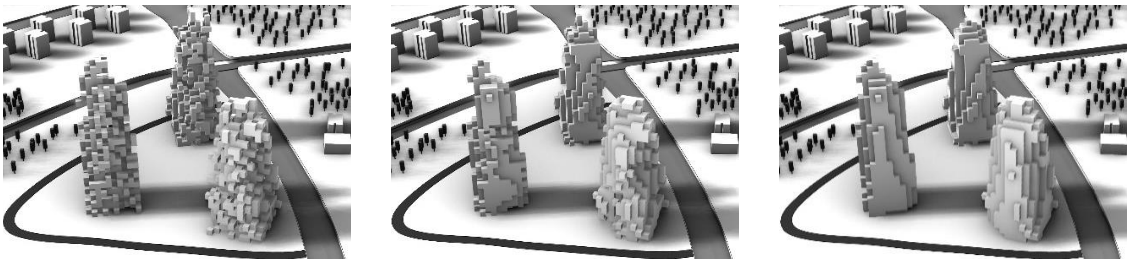
(g) [R=4; B_C=6*6*5; S=72; G_iS= 3; D_eC= 60%]

(h) [R=4; B_C=10*10*5; S=72; G_iS= 3; D_eC= 60%]

Fig. 11 Different models created in the third step

In this study, for each cell, dimensions equal to 3 m × 3 m and form porosity equal to 60% of the total volume were considered. The form of the hidden shell, which can be considered different according to the opinion of the designer, caused the final cells to become porous cells in the space around the volume and continuous cells in the inner space. By overlapping the cells and using the perimeter lines of each cell, a suitable space can be generated to create architectural space.

Finally, after performing three consecutive steps and in total after creating 10,000 different optimization states in the first and second step and more than 100 different states in the third step,



[R=4; B_C=5*5*5; G_iS= 0; D_eC= 40%] [R=4; B_C=5*5*5; G_iS= 3; D_eC= 40%] [R=4; B_C=5*5*5; G_iS= 9; D_eC= 40%]

Fig. 12 Cellular automata's progress

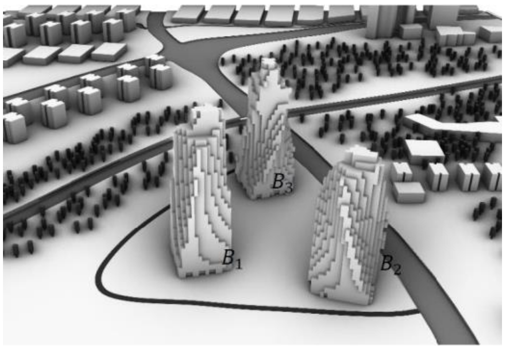
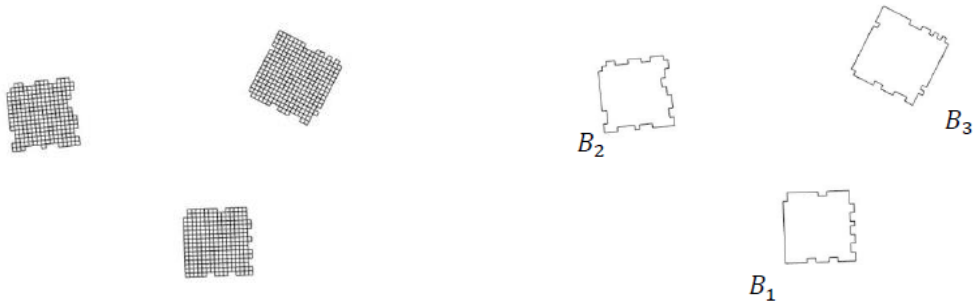


Fig. 13 The final model selected *B_1=[R=4; B_C=3*3*5; S=65; G_iS= 9; D_eC= 60%] *B_2=[R=4; B_C=3*3*5; S=25; G_iS= 9; D_eC= 60%] *B_3=[R=4; B_C=3*3*5; S=55; G_iS= 9; D_eC= 60%]



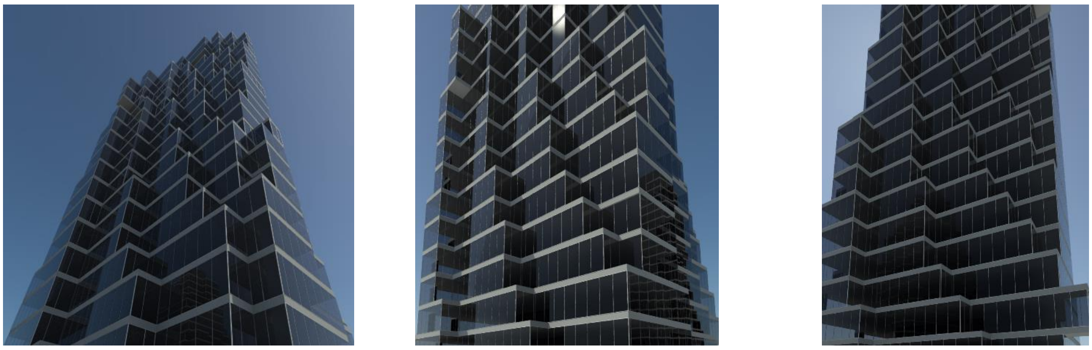
(a) Basic forms of plans (b) integrated form of plans

Fig. 14 Designable plan area

various forms were created. These forms are affected by cell density, the initial state of cellular automata, cell dimensions, and their generations (Fig. 11).

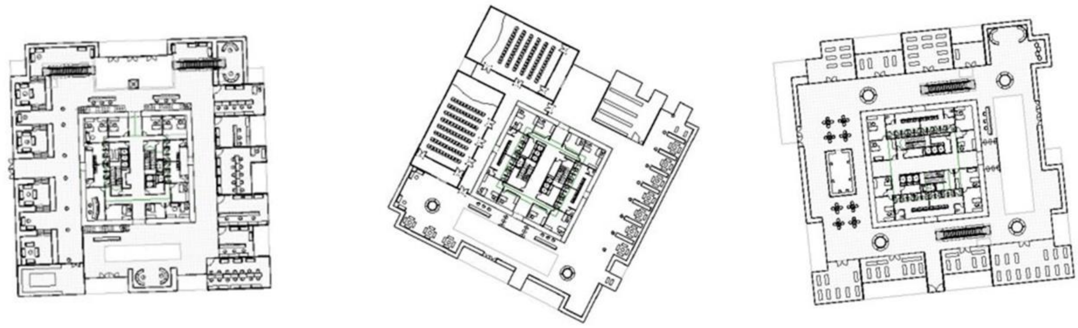
Fig. 12 also shows an example of cellular automata's progress which happens in the process of form-finding. Because this process occurs slowly, in order to show changes clearly, we had to choose forms after a few generations.

Among various states obtained in this research and to complete the design process, the design team decided to use three states with different specifications and appropriate to the use of the building. Fig. 13 presents the final form considered for this research. At this stage, with the entry



(a) (b) (c)

Fig. 15 Perspective of forms



(a) Plan of building B_1 (b) Plan of building B_2 (c) plan of building B_3

Fig. 16 sample plans of buildings

of the designer, it will be possible to complete the design process. The final answers, both in terms of form (Fig. 11) and plan (Fig. 14), have the required ability to become an architectural space such that the designer can design the plan in the spaces according to his opinion and taste.

At the last step, to evaluate the efficiency of design process and generated forms in architecture, we converted final forms to architectural forms in Revit software. As you see in Figs. 15(a)-15(c), volumes created are suitable for our needs in architecture and the architect can use of them by some simple changes. To prove the buildings have ability for planning, we also designed a sample plan for one floor of each buildings, as you see in Figs. 16(a)-16(c).

5. Conclusions

Although cellular automata has many advantages for using in architecture, such as following the rule of “form follows function”, making 3D forms, generating various forms and so on, use of it has been encountering many problems.

The contribution of this article relies on the process that we created. As we mentioned in previous sections, many researchers have studied to use CA in architectural design. The two main problems they have faced are; 1-the lack of relation between CA’s discrete space and Architectural continuous spaces, and 2- Inadequate forms produced with the needs of architecture. They tried to solve these problems in many ways; however, some solutions did not work correctly. For example, Krawczyk (2002) and Herr *et al*. (2015) modified the shapes and cells' dimensions to achieve an appropriate architectural form. Herr *et al*. (2015) also defined new roles based on their architectural purpose, but at the final step, she admitted the forms created by new roles are not suitable for architecture. Araghi and Stouffs (2015) also use CA in other way. They considered two criteria, access to core and lighting, to generate architectural forms by using CA. They specify some cores, and based on cells were removed or obtained based on their accessibility to cores. As you see, most of these articles used CA alone and cannot respond to some of the architectural requirements.

In this manuscript, to address the problems, we propose a hybrid method that includes three steps based on the combination of parametric design and genetic algorithm to design an optimized volumetric shell at the first two steps and the combination of cellular automata with this volumetric shell at the third step. Based on this process, we succeeded in making a relation between discrete CA's space and continuous architectural space and achieving various CA's forms that we are sure they are optimized and meet all our architectural requirements because of following our optimized volumetric shells.

Finally, it is noteworthy that although any change in the final solution will cause the optimization proposed in the early stages of the algorithm to be ineffective, the architect is allowed to make small amendments to the final form based on his opinion. These changes should be limited so as not to disrupt the connections made in the pores of the building forms and the final form.As we mentioned the purpose of involving the architect in the process is to achieve better solution and solve possible problems during the design process.

References

Araghi, S.K. and Stouffs, R. (2015), "Exploring cellular automata for high density residential building form generation", *Automat. Constr.*, **49**, 152-162. https://doi.org/10.1016/j.autcon.2014.10.007.

Benjamin, D., Stoddart, J., Nagy, D. and Villaggi, L. (2019), "Generative space planning in architecture design for efficient design space exploration", *U.S. Patent Application*, No.15/956,689.

Caetano, I., Santos, L. and Leitão, A. (2020), "Computational design in architecture: Defining parametric, generative, and algorithmic design", *Front. Arch. Res.*, **9**(2), 287-300. https://doi.org/10.1016/j.foar.2019.12.008.

Cerruti, U., Dutto, S. and Murru, N. "A symbiosis between cellular automata and genetic algorithms", *Chaos Soliton Fract.*, **134**, 109719. https://doi.org/10.1016/j.chaos.2020.109719

Coates, P., Healy, N., Lamb, C. and Voon, W.L. (1996), "The use of cellular automata to explore bottom up architectonic rules", *Proceedings of the Eurographics UK Chapter 14th Annual Conference*, 26-28, London, U.K., March.

Cortés, H., Tovar, A., Munoz, J.D., Patel, N.M. and Renaud, J.E. (2005), "Topology optimization of truss structures using cellular automata with accelerated simultaneous analysis and design", *Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization (WCSMO6)*, Rio de Janeiro, Brazil, May.

Cruz, C., Karakiewicz, J. and Kirley, M. (2016)," Towards the implementation of a composite Cellular Automata Model for the Exploration of Design space", *Proceedings of the 21th International Conference of The Association for Computer-Aided Architectural Design Research in Asia*, 187-196, Melbourne, Australia, March.

Cruz, C., Kirley, M. and Karakiewicz, J. (2017) "Generation and exploration of the architectural form using a composite cellular automata", *Proceedings of the Australian Conference on Artificial Life and Computational Intelligence*, 99-110, Australia, January. https://doi.org/10.1007/978-3-319-51691-2_9.

Devetakovic, M., Petrusevski, L., Dabic, M. and Mitrovic, B. (2009), "Les folies cellularies- an exploration in architectural design using cellular automta", *Proceedings of the 12th Generative Art conference*, 181-192, Italy, December.

Dijkstra, J., Jessurun, J. and Timmermans, H.J. (2001), "A multi-agent cellular automata model of pedestrian movement", *Pedestrian Evacuat. Dynam.*, **173**, 173-180.

Dincer, A.E., Cagdas, G. and Tong, H. (2014), "A digital tool for customized mass housing design", *Proceedings of the 32th eCAADe Conference*, **1**, 201-211, England, U.K., September.

Dounas, T., Spaeth, B., Wu, H. and Zhang, C. (2017), "Dense urban typologies and the game of life: evolving cellular automata", *Proceedings of the 17th International Conference on Computer Aided Architectural Design Futures*, Istanbul, Turkey, July.

Fathy, F., Mansour, Y., Sabry, H., Abdelmohsen, S. and Wagdy, A. (2015), "Cellular automata for efficient daylight performance: optimize façade treatment", *Proceedings of BS2015: 14th Conference of International Building performance Simulation Association*, 2705-2711, India.

Funes, P.J. (2001), *Evolution of Complexity in Real-World Domains*, Brandeis University, Massachusetts, U.S.A.

Gunawan, A.Y., Harapan, R., Ihsan, A.F. and Soetikno, D. (2017), "Simulation of gas dispersion arising from gas venting process using cellular automata", *Proceedings of the 12th International Conference on Civil Environmental Engineering*, Bali, Indonesia, October. https://doi.org/10.17758/EIRAI.P0101716.

Hajela, P. and Kim, B. (2001), "On the use of energy minimization for CA based analysis in elasticity", *Struct. Multidiscipl. Optim.*, **23**(1), 24-33. https://doi.org/10.1007/s00158-001-0162-2.

Herr, C.M. and Ford, R.C. (2015), "Adapting cellular automata as architectural design tools", *Proceeding of the 20th International Conference of the Association for computer-Aided Architectural Design Research in Asia*, 169-178, Daegu, South Korea, May.

Herr, C.M. and Kvan, T. (2005), *Using Cellular Automata to Generate High-Density Building Form* in *Computer Aided Architectural Design Futures*, Springer, Dordrecht, Netherlands. https://doi.org/10.1007/1-4020-3698-1_23.

Herr, C.M. and Kvan, T. (2007), "Adapting cellular automata to support the architectural design process", *Automat. Constr.*, **16**(1), 61-69, https://doi.org/10.1016/j.autcon.2005.10.005.

Kicinger, R., Arciszewski, T. and De Jong, K. (2004), "Morphogenesis and structural design: Cellular automata representations of steel structure in tall buildings", *Proceedings of the 2004 Congress on Evolutionary Computation*, **1**, 411-418, Portland, U.S.A., June.

Krawczyk, R.J. (2002), "Architectural interpretation of cellular automata", *Proceedings of the International Conference on Generative Art*, **1**.

Lee, Y. and Kim, S.H. (2016), "Algorithmic design paradigm utilizing cellular automata for the han-ok", *Nexus Network J.* **18**(2), 481-503. https://do Ahmet i.org/10.1007/s00004-016-0292-x.

Marenko, A. (2019), "Presumable application of cellular automata with strong anticipation in quantum physics", *J. Phys.*, **1251**(1), 01203. https://doi.org/10.1088/1742-6596/1251/1/012031.

Menshutina, N.V., Kolnoochenko, A.V. and Lebedev, E.A. (2020), "Cellular automata in chemistry and chemical engineering", *Ann. Rev. Chem. Biomol. Eng*, **11**, 87-108. https://doi.org/10.1146/annurev-chembioeng-093019-075250

Parmee, I. (2004), *In Adaptive Computing in Design and Manufacture VI*, Springer, Dordrecht, Netherlands.

Pelechano, N. and Malkawi, A. (2008), "Evacuation Simulation models: Challenges in modeling high rise building evacuation with cellular automata approach", *Automat. Constr.*, **17**(4), 377-385. https://doi.org/10.1016/j.autcon.2007.06.005.

Rahbar, M. (2018), "Application of artificial intelligence in architectural generative design", *eDAADe*, **1**, 71-72.

Rahbar, M., Mahdavinejad, M., Markazi, A.H. and Bemanian, M. (2022), "Architectural layout design through deep learning and Agent-based modeling: A hybrid approach", *J. Build. Eng.*, **47**, 103822. https://doi.org/10.1016/j.jobe.2021.103822.

Shiffman, D. (2012), *The Nature of Code*, U.K.
Slotta, D.J. (2001), "Structural design using cellular automata", Ph.D. Thesis, Virginia Tech, Verginia, U.S.A.
Von Neumann, J. and Burks, A.W. (1966), "The theory of self-reproducing automata", *IEEE T. Neural Netw.*, **5**(1), 3-14.
Wolfram, S. (2002), *A New Kind of Science*, Wolfram Media, U.S.A.
Zawidzki, M. (2010), "A cellular automaton controlled shading for a building façade", *Proceedings of the International Conference on Cellular Automata*, 365-372, Springer, Berlin, Germany. https://doi.org/10.1007/978-3-642-15979-4_38.
Zeng, J., Qian, Y., Mi, P., Zhang, C., Yin, F., Zhu, L. and Xu, D. (2021), "Freeway traffic flow cellular automata model based on mean velocity feedback", *Physica A*, **256**, 125387. https://doi.org/10.1016/j.physa.2020.125387.

CC