**Automated floor plan generation by a combination of evolutionary algorithm (genetics) and machine learning (k-nearest neighbors and k-means clustering**)

Reza babakhani 1\*, Researcher PhD of Architecture, Department of Architecture, Science and Research Branch, Islamic Azad University, Tehran, Iran, [reza.babakhani@srbiau.ac.ir](mailto:reza.babakhani@srbiau.ac.ir)

Mahsa safarnejad 2, Researcher PhD of Architecture, Department of Architecture, Science and Research Branch, Islamic Azad University, Tehran, Iran, Mahsa.safarnejad@gmail.com

**Automated floor plan generation by a combination of evolutionary algorithm (genetics) and machine learning (k-nearest neighbors and k-means clustering**)

Abstract

Design is a fundamental, problem-oriented, purposeful, and comprehensive activity, and, more than three decades after the advent of computers in the process of implementing architectural designs, the design process is still done by humans on paper versions and, then transferred to the computers by the user through the software, in which the computers have no inherent design intuition, this is the main issue of delegating the design process to computers.

This research aims to find a new method for combining artificial intelligence algorithms for the automatic production of architectural plans that can identify the appropriate design based on user needs and simultaneously apply rules, regulations, and design standards. But the research hypothesis is that by using evolutionary algorithms and machine learning interactively, a process can be created to achieve relative intuition in machines.

The research method in this research is of the library, documentation, and quantitative data analysis type along with the use of (genetic) algorithms, supervisory learning algorithms, and Python libraries. Findings show that the use of feature vectors as supervised learning can lead to the identification of the best design and provide relative intuition in machines. Also, the use of a genetic algorithm to search the design space and equalize the plans based on the dimensions of the user's land would be a good suggestion. Finally, by saving the design process experience by algorithms each time, it is possible to provide a context for reinforcing learning.

Automatic Plan Generation, Genetic Algorithm, Machine Learning

**1. Introduction**

Computer-aided design (CAD) software has had a dramatic impact on architectural practice since the emergence of computers in academia in the 1950s, and especially since the introduction of personal computing in the1980s. Although early researchers envisioned a wide-ranging future interaction between computers and human designers [1], the first computer tools to be widely adopted by architectural designers were computerized versions of traditional drafting and rendering tools. While they allowed designers to produce content much faster than with traditional methods, they did not fundamentally change the process of design [2].

The concept of generative design, as described in this paper, addresses this limitation by tasking a computer to explore a design space semi-autonomously, and then report back to the designer which options it considers promising for further analysis. Because a computer can process information much quicker than a human, such a system allows a much deeper exploration of complex design spaces. Traditionally, such an approach has been used to optimize a given model to achieve maximum possible performance based on concrete objectives [3].

Computer does not have any inherent intuition about design and the human designer must explicitly describe to the computer how to determine which designs perform better than others. The model needs to be connected to a search algorithm that can control the input parameters of the model, get feedback from the metrics, and intelligently tune the parameters to find high performing designs while also exploring the full possibilities of the design space. One of the most promising of these algorithms is the multi-objective genetic algorithm (MOGA) which uses principles of evolution to create sequential generations of designs and evolve them to contain higher performing designs over time [4].

The quantification of spatial experience has also been explored by a variety of authors. Hillier et al. [5] proposed a variety of analytical tools for studying spatial configurations which they called ‘space syntax’. Peponis et al.[6] extended this work by proposing a universal method for understanding plan topology through linear representation.[7]Design is a fundamental, purposeful, pervasive, and ubiquitous activity and can be defined as the process of creating new structures characterized by new parameters, aimed at satisfying predefined technical requirements. It consists of several phases, which differ in details such as the depth of design, kind of input data, design strategy, procedures, methodology, and results [8].

Goldberg presents an idealized framework for conceptual design in four components: problem, designer, alternative designs and design competition, and shows how evolutionary techniques (specifically genetic algorithms) can be thought of as ‘a lower bound on the performance of a designer that uses recombinative and selective processes [9]. Rosenman has explored evolutionary models for non-routine designs [10] and has investigated the generation of creative house plans (later referred to as floorplans in this paper) using genetic algorithms [11].

Creation of floorplans has also been investigated by Gero and Schnier as an evolving representation problem that restructures the search space in [12] by co-evolution of design and solution spaces in [13] and using case-based reasoning by De Silva Garza and Maher in [14]. At the same time, collaborative systems have been the focus of studies into creativity and computer supported cooperative work [15] since the early 90s. There has been a paradigm shift from computer-aided design systems to computer supported collaborative design systems [16]. It has been argued that much of our intelligence and creativity results from interspaces. We present a collaborative interactive genetic algorithm implementation for our model to evolve floorplans and widget layout/style design, as a user-interface development tool [17]. Most approaches probe possible placements in a design space. Galle et al. [18] implemented an exhaustive algorithm to select the rectangular arrangements satisfying constraints among all the possible generations. However, due to computational restrictions, the method could only handle layouts with up to ten rooms [19].

Evolutionary algorithms have been applied to search layout possibilities by treating design variants via crossovers and mutations operators in these approaches. Evolutionary strategy is used to fit rooms into target envelopes, while improving appropriately designed fitness functions. Furthermore, mutations are allowed during such evolution, for example to switch rooms (crossover) in order to optimize connectivity [20]. Recently, an automated layout generation has been proposed to sample and efficiently explore the layout search space [21]. The method, however, is not designed to support interactive design refinements. In a related attempt, Harada et al. [22] designed a system that allows users to interactively drag rooms, but the possible layouts are predefined. Specifically, the algorithm searches for a matched state that best reflects user intents from a set of constructed transformations for mapping states. Moreover, only limited sets of constraints are considered, and the method does not generalize to handle manufacturing constraints. More recently, physical and manufacturing considerations have also been explored in the context of geometric form finding [23]. Similarly, in this work, we focus on pre-cast concrete-based constructions and consider its implications in design and layout problems.

The most common genre of FLP involves a finite number of rectangular building blocks or modules 𝑀𝑖 (𝑖 = 1, 2 … 𝑁), representing various activities or functional units such as departments, machines, rooms, cells, activities, or spaces. The objective is to minimize the cost of inter-module flow by placing all the modules on the packing space without overlaps, in such a way that the edges of 𝑀𝑖 are parallel to the x and y axes respectively. It is a well-known NP-complete problem; thus, a verifiably optimal solution cannot be known even for modest-size problems [24]. The purpose is usually to minimize costs, time, or distance in the flow of material and occupants through different departments [25].

The methodologies vary from heuristic search methods [26] and genetic algorithms [27] to mixed-integer programming [28] and threshold-accepting algorithms [29]. These automated design techniques caught the attention of researchers from the field of architecture as they could potentially solve the problem of large spaces associated with hospitals and schools [30].

Doulgerakis [31] used genetic programming with an agent-based approach to assign activities (space functions) to geometric rooms. In this case, the vertical circulation, and their consequent implications were ignored. Flack [32] Dolgrakis tested two methods of evolutionary computation, genetic algorithm, and genetic programming, and in multi-story buildings, proposed solutions to place the staircase as a fixed space in the problem, which is repeated at each level.

There was no need to change the form and move the floors and this element was drawn separately in the plan. Beyond the issue of floor connections, Zimmerman used a rectangular partitioning method in which restrictions on several specific levels were applied, such as vertical wall alignments or recessed space constraints on walls and walls [33]. This method was also considered for a long time in order to complete the previous methods.

But over time, the goal of productive and computer-aided design is to plan living space based on the placement of objects in regular or irregular shapes by software in a specific architectural design by a designer, and this has become one of the most popular design methods. This is because of the widespread use in people's daily lives, such as placing books on bookshelves [34], placing cars in the car park [35], and placing objects or pictures on PowerPoint slides, which can be used for All should be applied equally, and solving this problem is one of the most fascinating and challenging problems that researchers are looking for a solution to [36].

The production of architectural designs requires spatial planning and the goal is to find practical places and dimensions for a set of interconnected objects that meet all design requirements and have the maximum design quality in terms of design preferences. [37] . This is the need of architects and society because they have to come up with acceptable designs instead of non-optimal solutions. Computer technology was used in the mid-1960s for the structural implementation of architectural designs [38]. Now, with advances in computational capabilities over the years and algorithm modifications, it seeks to solve such design problems and find practical solutions to minimize human interference in the design process, along with the dual challenge of addressing constraints. Topological and dimensional properties of spaces [39].

It is going back and forth the topological constraints control a set of spaces and the relationship between them and make them responsive to each other in order of spatial arrangement, while the dimensional constraints applied to space are dimensions that are possible for a particular space. Different researchers and architects have given different preferences to two sets of constraints and have considered different priorities in the automated design process [40].

In fact, by examining the theoretical foundations of the previous research, the problem extracted is that the process of designing architectural plans is not a task which can only be done by machine and algorithms and also all the effective points and data can't be calculated and applied by the designer, rather, a method must be innovated that combines the original design by the designer with the optimization, plotting, and application of rules based on user needs by algorithms.

The main objective of this research is to find a solution that can minimize the human interventions in the optimization process and achieve an ideal design and that, the primary design is somehow done from human studies and activities and ideas, and not like putting a book on a shelf, in fact, the goal is to achieve an appropriate method of automatic architectural plans production by artificial intelligence with the ability of identifying the appropriate design based on user needs. At the same time, it can provide rules, regulations and design standards based on the dimensions and sizes of the designed land.

The research hypothesis now is that a process can be created by using evolutionary algorithms (genetics) and machine learning algorithms (k-means clustering) and (k-nearest neighbors) interactively and simultaneously so that this objective (automatic architectural plans production by artificial intelligence) is realized.

**2. Background research**

According to Table 1, in 1976, Mitchell proposed the first theory of synthesis and optimization in the generation of architectural plans, where sixteen possible positions of squares and rectangles were achieved. Also, in 1996, Jo and Gero simulated architectural plans based on a study using genetic algorithms. In 1997, researchers such as Rosenman, Schnier and Gero, Gero and Kazakov, and Jagielski and Gero were able to generate a sample of architectural plans using genetic algorithms and genetic programming.

From 1998 to 1999, Garza and Maher, Bentley, Elezkurtaj and Franck, in a study using genetic algorithms and evolutionary strategy algorithms, were able to create architectural spatial combinations. But following research from 2001 to 2002, for the first time, they were able to generate architectural plans through genetic algorithms, sequential quadratic programming, and reproduction of simulations, in addition, from 2003 to 2008, attempts were made to generate architectural plans with genetic algorithms and genetic programming, led by Makris, Virirakis, Makris, Bausys and Pankrasovait, Homayouni, Doulgerakis, Banerjee et al., and Serag et al.

However, from 2009 to 2011, there were changes in the research of this paradigm and from the use of single genetic algorithm or programming toward the combination of genetic algorithms with other emerging algorithms. Researchers such as Inoue and Takagi, Wong and Chan, Benjamin Dillenburger, Thakur et al., de la Barrera Poblete, Knecht, and Flack developed different architectural plans. Also, from 2011 to 2012, Ricardo Lopes et al. and Reinhard Koenig were able to use spatial divisions to generate new plans through a hierarchical algorithm method.

**Table1**. Background of plan generation research [41].

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Year | oF | wD | eD | eW | iD | S | fL | eF | bB | aB | oO | sA | Method |  | Researcher |  |
| 2010 | gts |  | • |  | • |  |  |  |  |  |  | • | GA/DA |  | Thakur et al |  |
| 2010 | g |  |  |  |  | • |  |  |  |  |  |  | GA+VD |  | de la Barrera Poblete |  |
| 2010 | gt |  |  |  |  | • |  |  | • |  |  |  | GA/ES+K-D |  | Knecht |  |
| 2011 | gt |  |  |  |  | • | • |  | • |  |  | • | GA/GP |  | Flack |  |
| 2011 |  |  |  |  |  |  |  |  |  |  |  |  | AH |  | Ricardo Lopes et al |  |
| 2012 |  |  |  |  |  |  |  |  |  |  |  |  | AH |  | Reinhard Koenig |  |
| 2012 | gt | • | • | • | • | • |  |  | • | • | • | • | ES+SHC |  | Rodrigues |  |
| 2015 |  | • | • |  |  | • | • |  | • |  |  | • | GA/GP |  | Victor Calixto |  |
| 2019 |  |  |  |  |  | • | • |  | • |  |  | • | GAN |  | Stanislas Chaillou |  |
| 2020 |  |  | • |  |  | • | • |  | • |  |  | • | Graph2Plan |  | RUIZHEN HU |  |
| 2020 |  | • | • |  |  | • | • |  | • |  |  | • | GA |  | Maciej Nisztuk |  |

GA: Graph algorithm,SO: Synthesis and Optimization, GA: Genetic Algorithm, GP: Genetic Programming, ES: Evolutionary Strategy, SA: Simulated Annealing, SQP: Sequential Quadratic Programming, L: Lindenmayer System, VD: Vernoy Diagram, DA: Digestra Algorithm, SHC : Search of Hill Climbing, AH: Hierarchical Algorithm, Graph2Plan: Pixel to Pixel, t: Topological, h: Heating, c: Cooling, l: Lighting, s: Walking Distance, oF: Objective Function, wD: Wall Dimensions, eD: External Door, eW: Window, iD: Internal Door, S: Spaces, fL: Floors, st: Stairs, eL: Elevators, eF: ​​Equipment- Furniture, bB: Building Area, aB: Adjacent Buildings, oO: Openings orientation, sL: Location, sA: Adjacent spaces.

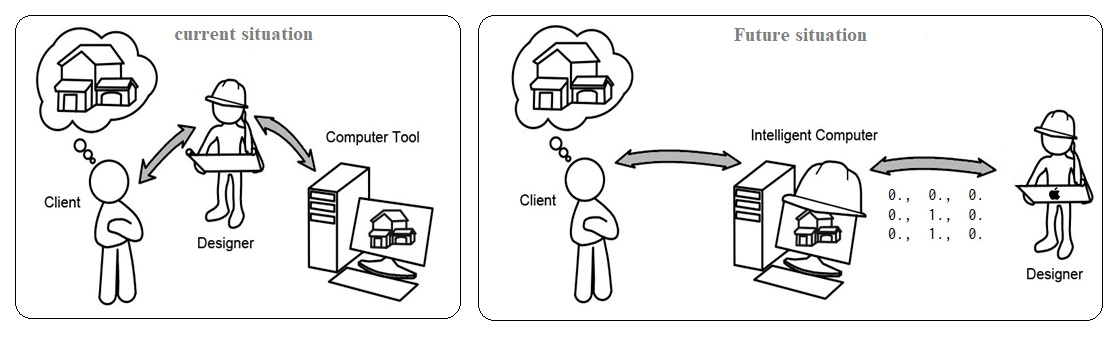
The method column of Table 1 shows the research of the last 10 years, in which the move towards combining algorithms is a priority for the researchers and, there have been successes in the use of hybrid algorithms in the production of plans based on multi-factor design, the hypotheses of these studies have been based on the fact that by combining different artificial intelligence algorithms (genetic algorithm, optical hill, swarm) while speeding up the plan production process, various environmental data, energy issues and design criteria can be included in the design with appropriate quality.

But in all previous research, most algorithms have been combined within their categories, such as combining genetic algorithms with genetic programming or hill climbing algorithms with genetic programming, which aims to use evolutionary algorithms and has been tested several times and, has been tried to discover the design space with genetic algorithms or hill climbing and to suggest a suitable design for it, but it has failed. And each time the algorithm could not extract more than 10 spaces regularly and suffered harmful mutations in the design process on the other hand, these mutations are in the structure of evolutionary algorithms, such as genetic mutations in nature, which are sometimes useful and sometimes harmful and, the second problem is that these algorithms need a lot of time to search the design space.

The second group includes the deep learning algorithms that we still combine algorithms with; we do not see evolutionary algorithms, these conflicting GAN algorithms have become common in recent years and have been simulated as architectural plans based on the pixels of the plan images or the placement of color pixels with a specific user concept in the vicinity of the same spaces. But this method has drawbacks such as a lack of scalability and because the unit and method of work are based on pixels of plan images, while the architecture has a scale based on meters and centimeters. Also, there is no ability to implement optimization processes in these plan images on the other hand, the issues and regulations of the National Building Regulations cannot be interpreted in these pixels. Therefore, it seems that this method is more used to produce the primary design or basic ideas, but in principle, there is still a difference of opinion and taste.

By studying the results of previous research, it is possible to know the difference between this research and other research in the point that this research for the first time follows the combination of evolutionary algorithm (genetic) with the structure of supervised learning algorithms (k-means clustering). Also, the second distinction of this research is the introduction of a method based on the use of input data as a numerical vector by the user to identify the user's demands by the algorithm and design based on environmental standards and conditions.

A review of the research background shows that architecture is not a process that can be mechanized by eliminating the architect. So that the architect no longer needs to be involved, but the presence of the architect in the automated design process is inevitable. But as a teacher and professor who can train the architectural data like the examples of correct plans to the machine (based on Figure 1) and, then the algorithm can match it based on climatic conditions, user needs, urban planning rules and regulations, and finally plot it in architectural software and send it to the user.



**Figure 1**. Changing the position and role of the architect in the design process.

**3. Methodology**

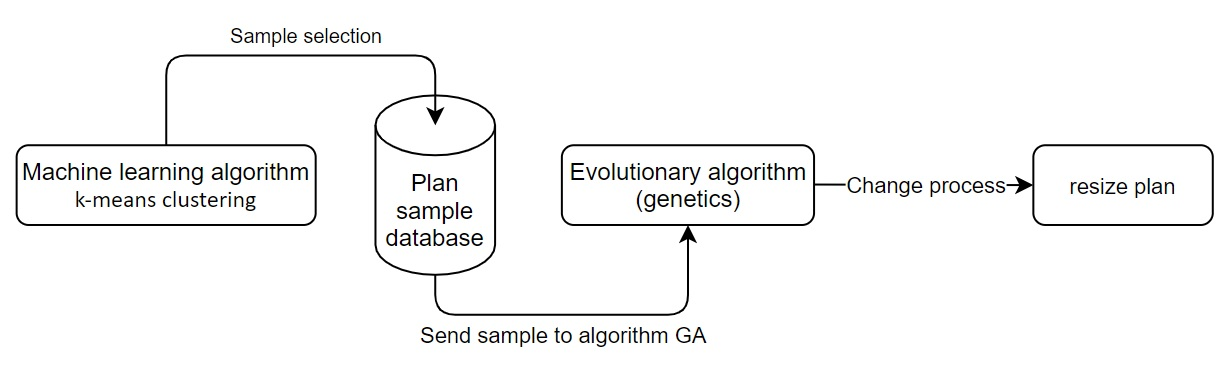
The proposed method is based on evolutionary algorithms (genetics) and machine learning (KNN and k-means clustering) to generate architectural plans called MLGAFPG, which is proposed to allocate space in a two-dimensional level for common urban lands. The MLGAFPG consists of two techniques, the first based on machine learning and the second on genetic algorithm. In the machine learning algorithm, the input information is entered by the user, and samples of architectural plans based on machine-supervised learning have already been learned.

The algorithm proposes designing a considered plan among several thousand learned models based on the user’s need and land data. Here, these samples are collected based on the most common dimensions of urban lands and completed according to architectural standards. Then, through the second algorithm, as genetic evolutionary algorithms and according to the specialized and professional preferences and limitations of architecture, by an evolutionary strategy (Genetic), the production process and matching the floor plans of the architectural floors begin with the dimensions of the user's mainland.

This is a special two-step approach that can be used to produce architectural plans. In fact, in the first stage, architectural plans are trained to the machine learning algorithm as learnable examples, and then in the next stage, with the help of GA, in the process of several repetitions, it changes the geometric form and brings to the main coordinates of designing land. The aim of the GA stage is a local search for the most suitable position of the architectural plan spaces in the selected land based on common urban examples for design. With the help of GA, the best state is maintained in the stage of searching algorithm generation space, which is equal to the coordinate data of the spaces. GA is designed to search the areas among the sample spaces generated in the machine learning algorithm and the main dimensions are entered as land data.

The proposed method of automated production of architectural plans generates a process in which a set of objects (spaces / rooms, exterior doors and windows and interior doors) are drawn on a two-dimensional plane, a method in which topological relations, both geometric constraints and user requirements are satisfied. In fact, each floor map design produced is a set of plans learned by the machine learning algorithm and combined with user input information in a hierarchical manner. Each plan floor is made up of different spaces and elements, including the exterior windows, exterior and interior doors, and the floor of stages. This method can solve the main problem of mutation in the single evolutionary algorithms that results in the production of random spaces in more than 8 or 10 spaces.

Figure 2 shows a part of the process of plan selection and resizing it to the dimensions of the land intended by the user, here, first, the machine learning algorithm, based on the features demanded by the user, which have been presented in Table 2 of these features, has selected the appropriate sample from the database of the primary plans taught to artificial intelligence and then, it equalizes the dimensions of the sample with the main dimensions of the user's land with the help of genetic algorithm. Of course, other conditions such as; rules and regulations of design, energy optimization, economic cost calculations, energy production with solar systems, modulation of materials and several other capabilities can be implemented on the primary design plan. In future research, its methods will be discussed that, in this research only refers to the discussion of sample selection and its equailization.

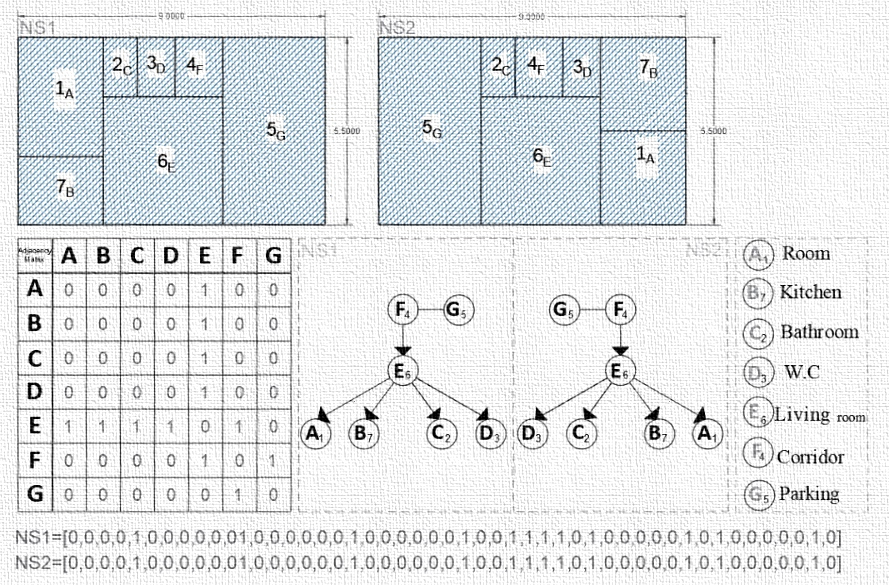


**Figure 2**. The process of changing the dimensions of the sample plan for the land proposed by the user.

**4. Discussion**

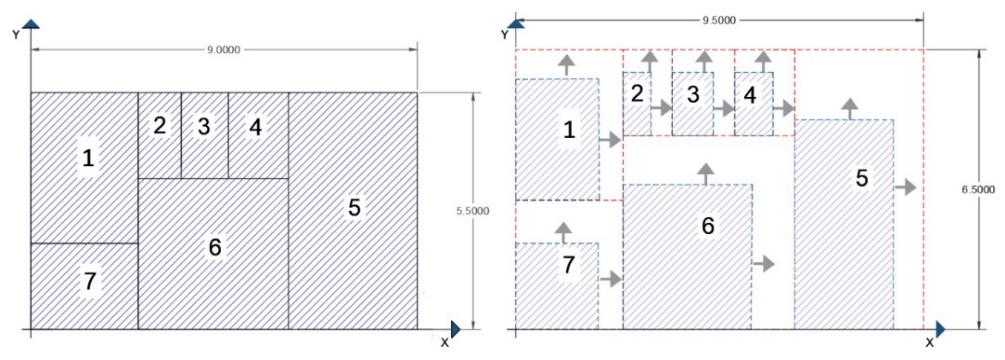
The need for interaction between machine learning algorithms and genetics algorithms is to build the automatic design intelligence of residential plans in Tehran city based on user needs and architectural criteria. In this research, the method of this interaction is analyzed and examined. To achieve this goal, steps must first be taken and the architectural data, which is qualitative and quantitative, turned into numbers.

In Figure 3, we can see an example of the spatial relationships of the architectural plan, which has been converted to zero and one codes, and the arrangement of these zeros and ones can be input for the algorithm to learn the spatial relationships in the plans numerically.

****

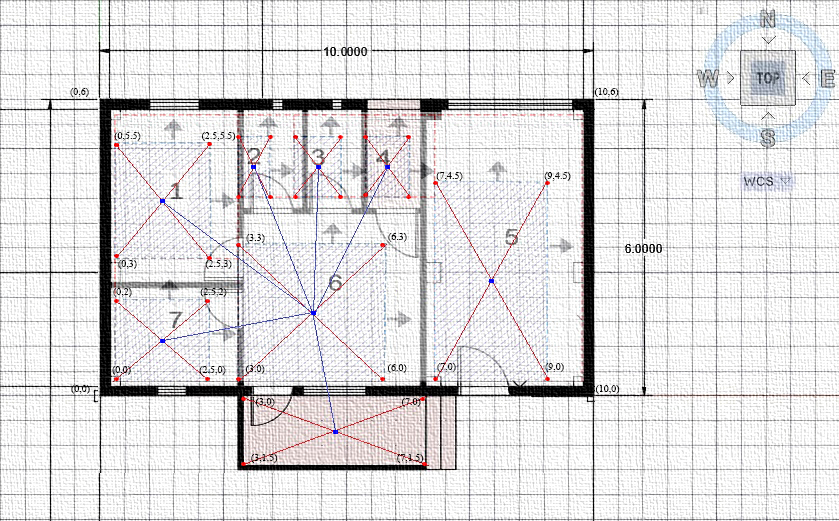
**Figure 3**. Construction of a numerical vector of the spatial relationships of architectural plans.

In the following, figure 4 shows how the genetic algorithm based on the search structure of the problem space can share the added area to the dimensions of the land among the available spaces vertically and horizontally and align the proposed plan for the user with it. In fact, in this algorithm, the goal is to achieve a plan where the user wants to receive a map based on the dimensions of his land and submit it to legal authorities, so the plan must comply with all designing rules and regulations and land dimensions.

****

**Figure 4**. Matching the dimensions of plan spaces by genetic algorithm with the dimensions of the main ground.

Figure 5 shows how the algorithm directs each space in the form of a square and rectangle with central control through rectangular points (x, y) in vertical and horizontal directions to match the added space to the dimensions of the trained plan by the machine to the main dimensions of the land in a two-way manner. Over time, this method classifies plans that have been generated with new dimensions, by non-supervised machine learning algorithms and adds in the memory of the algorithm’s automatic design intelligence as a new experience. These experiences will continue to be the basis for generating better plans by automated design intelligence algorithms.

****

**Figure 5**. Searching the design problem space using a genetic algorithm.

As mentioned before, in the former automated designing methods, only evolutionary algorithms have been used and all spaces, openings, and communication spaces have been found and applied by the search method in problem space; but in this research, for the first time, a combination of two machine learning algorithm and the genetic algorithm will be the basis for designing architectural plans. To achieve the automatic design intelligence of architectural plans, the features that make up an architectural plan are first extracted and categorized by data analysis algorithms, as illustrated in Table 2. These are some of the features that differentiate architectural plans from one another.

Here, in number 1, the user determines the design location, which is one of the cities, and in the second input, he enters the minimum and maximum width allowed for residential land, which is based on the most common dimensions built in the last 50 years.

Additionally, the third input is the virtual length that is selectable for the land; the fourth input is the land type which is grouped into four parts: north, south, east, and west; the fifth input is the type of building in terms of typical villa, multi-story villa and apartment; the sixth input is the type of windows that are classified according to the personality and interest of people into large, small, medium and floor to ceiling; the seventh input is the number of floors from one to five, which is adjusted based on the number of common floors; the eighth input is the number of units that the user can specify; the ninth input is the number of rooms that can be specified as input data from 1 to 5 rooms.

The tenth input is the number of resident population per unit, which is directly related to energy problems; the eleventh input is the parking lot orientation, which is on the right or left side of the building for left-hand and right-hand people; the twelfth input determines the location of the bedroom (Some people are interested in the bedrooms being in different directions and experiencing different lights).

The thirteenth input is the kitchen orientation, which is determined by the user based on people's interests and tastes, the fourteenth input is the kitchen model from the ordinary type to an island one; the fifteenth input is the bathroom model, which has several models; the sixteenth input is a corridor that allows people to have a corridor in the house or need uniform spaces without a partition. Finally, the seventeenth to twentieth inputs are based on national regulations. These cases can determine the type of plan that automatic design intelligence will design for the user and draw in AutoCAD software.

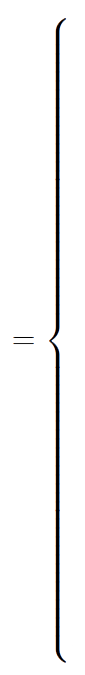
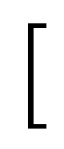
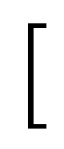
**Table 2**. Information on architectural plan features.

|  |  |  |
| --- | --- | --- |
| Row | User input information | User selection range |
| 1 | Location of the city | Tehran |
| 2 | Width of the ground | 7 to 20 |
| 3 | Length of the ground | 10 to 25 |
| 4 | Land type | North, South, Eastern, Western |
| 5 | Building type | Home, House, Apartment |
| 6 | Window type | Small, Medium, Large, Floor to ceiling window |
| 7 | Number of floors | 1 to 5 |
| 8 | Number of units | 1 to 3 |
| 9 | Number of rooms | 1 to 5 |
| 10 | Number of population | 1 to 5 |
| 11 | Parking place | Right or Left |
| 12 | Bedroom location | North, South, East, West |
| 13 | Kitchen location | North, South, East, West |
| 14 | Kitchen model | Closed, Open, Kitchen Island |
| 15 | Bathroom model | Bathroom master, Bathroom and toilet, Bathroom |
| 16 | Corridor | Yes or No |
| 17 | Stairs | According to the regulations |
| 18 | Elevator | According to the regulations |
| 19 | Light well | According to the regulations |
| 20 | Columns | According to the regulations |

To make it easier to write functions and function call in the problem statement process, subjects and inputs are abbreviated, and according to Table 3, each of the variables and input data is equivalent to a short abbreviation of the full subject. These data are then implemented as numerical data based on Figure 6 so that the machine can perceive and analyze them.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | USR | User selection range | *28* | HOM | Home |
| 2 | LC | Location of the city | *29* | HOU | House |
| 3 | WG | Width of the ground | *30* | AP | Apartment |
| 4 | LG | Length of the ground | *31* | SM | Small |
| 5 | LT | Land type | *32* | ME | Medium |
| 6 | BT | Building type | *33* | LA | Large |
| 7 | WT | Window type | *34* | FCW | Floor to ceiling window |
| 8 | NF | Number of floors | *35* | Ri | Right |
| 9 | NU | Number of units | *36* | Le | Left |
| 10 | NR | Number of rooms | *37* | CLO | Closed |
| 11 | NP | Number of population | *38* | OPE | Open |
| 12 | PP | Parking place | *39* | KI | Kitchen Island |
| 13 | BL | Bedroom location | *40* | BMAS | Bathroom master |
| 14 | KL | Kitchen location | *41* | BTO | Bathroom and toilet |
| 15 | KM | Kitchen model | *42* | BAT | Bathroom |
| 16 | BM | Bathroom model | *43* | UIIV | User input information vector |
| 17 | COR | Corridor | *44* | COREX | Exterior Corridor |
| 18 | ST | Stairs | *45* | SMat | Space matrix |
| 19 | EL | Elevator | *46* | SPdir | Space direction |
| 20 | LW | Light well | *47* | Midd | Minimum door dimensions |
| 21 | COL | Columns | *48* | Miwd | Minimum window dimensions |
| 22 | N | North | *49* | MifH | Minimum floor height |
| 23 | S | South | *50* | MisL | Minimum space length |
| 24 | E | Eastern | *51* | MisW | Minimum space width |
| 25 | W | Western | *52* | MiA | Minimum minimum area |
| 26 | Cal | Construction area limit | 53 | Gal | Gross area limit |
| 27 | Ewth | Exterior wall thickness | 54 | Iwth | Interior wall thickness |

**Table 3**. Nomenclature



0

1

Tehran

LC =

HOM

HOU

AP

HOM, HOU, AP

BT =

N,S,E,W

LT =

N

S

E

W

0

1

2

3



0

1

2

MLGAFPG



WG =

7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

7 to 20



10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25

LG =

10 to 25



NF =

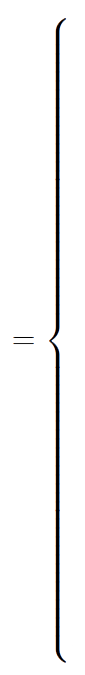
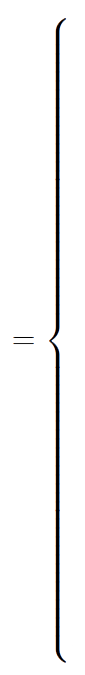
NU =

1 to 3

1,2,3

1,2,3,4,5

1 to 5



0

1

KL =

N,S,E,W

N,S,E,W

BL =

PP =

Ri Le

Ri

Le





0

1

2

3

N

S

E

W



1 to 5

NR =

1,2,3,4,5





NP =

1,2,3,4,5

1 to 5

0

1

2

3

N

S

E

W



MLGAFPG

MLGAFPG

SM

ME

LA

SM, ME, LA

WT =

0

1

2



0

1

CLO OPE

CLO ,OPE

KM =



COR =

Yes, No

Yes

No

0

1

0

1

2

11111100

BMAS

BTO

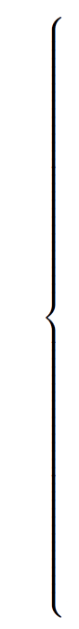
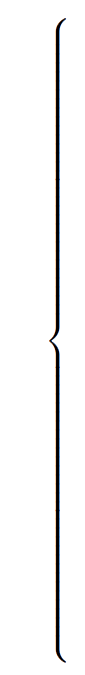
BAT

BMAS, BTO, BAT

BM =

**Figure 6.** View of the input detection algorithm and the design goal in automated design intelligence.

Based on the MLGAFPG algorithm, which is the automatic design intelligence of plans, it receives input data from the user and then converts it into encrypted data, which will contain various numbers. It then analyzes those numbers and converts encrypted data into a single cryptographic vector like Figure 7. Then, the resultant automatic design suits the user's needs, calculates it using the mathematical relation in Figure 8, and provides the best plan design state based on instructions, learning, and experience to the user.



0

1

0

10

20

0

1

0

0

2

1

1

1

2

2

1



0, 1, 0, 1, 0, 0, 1, 0, 0, 2, 1, 1, 1, 2, 2, 1

UIIV =

UIIV

**Figure 7**. Converting numerical cryptographers to machine unit vectors.

Based on Figure 6, the calculation of the similarity of the two data input vectors by the user and the data in the memory of automatic design intelligence learning begins here to reduce the complexity of the sample vector to two, the numbers 2,3,4,2 and 1, −2, 1,3 are converted and then based on the above relation, the analogy operations of vector A and vector B are applied by the relation xi-yi, which can be observed according to Figure 6 as (2-1)2 +(3 +2)2 + (4-1)2 +(2-3)2, the product of xi-yi subtraction and their exponentiation is the equation √ (1 + 25 + 9 +1) that if we apply the addition operators, we will encounter the number √ (36) and then if we get it out from the radical, we will reach 6, which is the similarity of the two input vectors to the above formula.

Based on the above formula of design intelligence, after recognizing the user's needs based on Figures 6, 7, and 8 and perceiving the spaces and the way they are placed in the architectural plan, one should be able to adjust it based on national standards, rules, and regulations and optimized by energy topics and climatic data must also be applied by automatic design intelligence in the designing process. Elaboration on each case is beyond the scope of this article, so in future research, rules and regulations and spatial layout will be further discussed.

Determine the Euclidean distance between u⃗ =(2,3,4,2) and v⃗ =(1,−2,1,3).

D (u⃗ ,v⃗ ) = ∥u⃗ − v⃗ ∥ =

D (u⃗ ,v⃗ ) = ∥u⃗ − v⃗ ∥ =

D (u⃗ ,v⃗ ) = ∥u⃗ − v⃗ ∥ =

D (u⃗ ,v⃗ ) = ∥u⃗ − v⃗ ∥ = 6

**Figure 8**. Formula for calculating each cryptographic vector with a similar vector in Cartesian coordinates.

According to Table 4, the minimum dimensions and sizes that must be observed in the design of architectural spaces have been extracted. It is stated that each space and element involved in architectural plans should have what dimensions, size, and orientation to be approved. It is also trained as shown in Figures 6, 7, and 8 and as vectors and numerical data to the automatic design intelligence of the plan, so that it can apply them in the designing process. Python programming language with its special libraries has been used to learn this data by the automatic design intelligence of architectural plans.

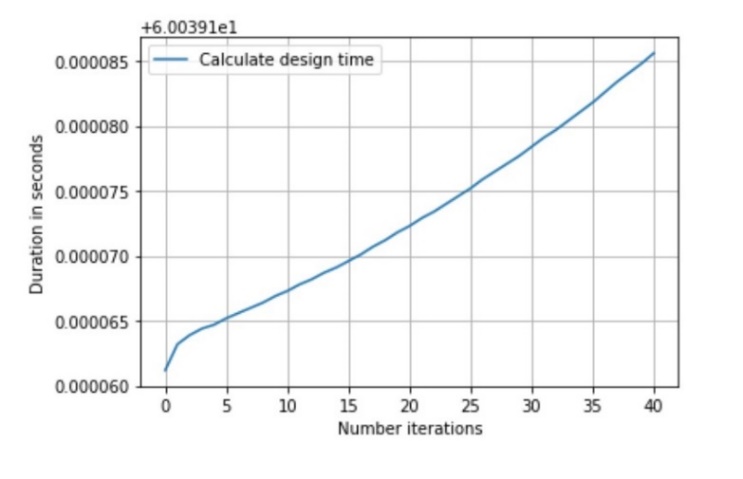
**Table 4**. Minimum rules and regulations that must be observed in the design of plans.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MiA | MisW | MisL | MifH | Miwd | Midd | SPdir | SMat | Name Space | Level |
| - | 1/40 | - | 3/24 | - | 1 m | - | 0 | Corridor Entrance | L1 |
| - | 1/10 | - | 3/24 | - | 1 m | - | 0 | Corridor |
| 12 | 2/70 | - | 3/24 | 1/8 | 90 cm | East | 1 | bedroom |
| 6/5 | 2/5 | - | 3/24 | 1/8 | 90 cm | East | 1 | bedroom |
| 12 | 2/70 | - | 3/24 | 1/8 | 1 m | South | 1 | Living room |
| 7/5 | 2/15 | - | 3/24 | 1/8 | 1 m | North | 2 | Kitchen |
| 12/5 | 2/5 | 5 | 2/88 | - | 3 m | - | 0 | Parking |
| - | 1/20 | - | 3/24 | 1/8 | 80 cm | West | 2 | Bathroom |
| 1/56 | 1/20 | 1/30 | 1/10 | - | 80 cm | East | 1 | Balcony |
| - | 1/10 | 1/20 | 3/24 | 1/8 | 80 cm | West | 2 | W.C |

After learning the standards and minimum criteria for design by automated design intelligence, now the spatial relationships of the plans and neighborhoods should be created by the spatial relations matrix of Figure 10, and this section should be simultaneous with the application of the standards along with equalization of the dimensions of the userland input that its infrastructure and build density, which has already been done in the data analysis section.

This starts the designing process. In fact, after receiving input information and converting it into cryptographs, and recognizing the user's needs automatic design intelligence tries to analyze information such as dimension, size, occupancy level, density, climate, and other basic variables. After analyzing, classifying, and converting them to cryptographic vectors, it designs and locates by referring to the training learned from different types of plans.

This process is similar to reading cryptographic vectors; Climate, criteria, and other design variables will continue, and at the same time, with the help of genetic algorithms, the dimensions of learning plans or automated design intelligence experiences will be closer to the dimensions of the standard occupancy level, to the extent that input dimensions and machine drawing dimensions to be the same. The whole process takes 50 to 100 seconds from the time the information is received by the algorithm to the time it is analyzed, grouped, and drawn in AutoCAD software and later stored and sent to the user according to Figure 9.



**Figure 9**. Calculation of analysis time and application of design process by automated design intelligence.

Based on Figure 10, In order for the automatic design intelligence to be able to draw the plans required by the user accurately and with the correct spatial relations, it is necessary to learn the spatial layout of the machine in the language of zero-one codes. For doing so, first a matrix from spatial layout is defined in the form of zero-one codes that if there is a connection between two spaces, the number one is used. If there is no number zero, and if there is a space inside another space with a partition door, the number two is used.



S1  S2  S 3 S 4 S 5 S 6 S 7 S 8

0 1 0 0 0 1 0 0

S1  S2  S 3 S 4 S 5 S 6 S 7 S 8

0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 0

0 0 0 0 0 1 0 1

1 0 1 1 0 0 1 0

0 0 0 0 0 0 0 0

0 0 0 0 0 1 0 0

0 0 0 0 0 1 0 0

1 0 0 0 0 0 0 0

0 0 0 0 0 1 0 0

0 0 0 0 0 1 0 0

0 0 0 0 0 0 1 0

0 0 0 0 0 1 0 1

0 1 1 0 1 0 1 0

0 0 0 0 0 1 0 0

0 0 0 0 1 0 0 0

S1 S2 S 3 S 4 S 5 S 6 S 7 S 8

S1 S2 S 3 S 4 S 5 S 6 S 7 S 8

Design 2=

Design 1=

0 0 0 0 0 1 0 0 0

0 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0



S1  S2  S 3 S 4 S 5 S 6 S 7 S 8

0 0 0 0 0 0 1 0

0 0 0 0 0 0 1 0

0 0 0 0 0 0 1 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 1 0

0 0 0 0 0 0 1 0

0 0 0 0 0 0 0 1 0

0 0 0 0 0 0 0 1 0

0 0 0 0 0 1 0 0 0

0 1 0 0 0 0 0 0 0

S1  S2  S 3 S 4 S 5 S 6 S 7 S 8 S9

0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 0 1 0

S1 S2 S 3 S 4 S 5 S 6 S 7 S 8 S9

Design 4=

S1 S2 S 3 S 4 S 5 S 6 S 7 S 8

Design 3=

0 0 0 0 0 0 1 0

0 0 0 1 1 1 1 0 1

0 1 1 1 1 1 0 1

0 0 1 0 0 0 0 1 0

0 0 0 0 0 0 0 1 0



0 0 0 0 0 0 1 0 0

0 1 0 0 0 0 1 0 0

1 0 1 1 1 0 0 1 0

0 0 0 0 0 0 0 0 0

S1  S2  S 3 S 4 S 5 S 6 S 7 S 8 S9

S1  S2  S 3 S 4 S 5 S 6 S 7 S 8 S9

0 0 0 0 1 0 1 1 0

0 0 0 1 0 0 0 0 0

0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 1 0 1

0 0 0 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0

0 0 0 0 0 1 0 0 0

Design 5=

Design 6=

S1 S2 S 3 S 4 S 5 S 6 S 7 S 8 S9

S1 S2 S 3 S 4 S 5 S 6 S 7 S 8 S9

0 1 0 0 0 0 0 1 0

0 0 0 0 0 0 0 1 0

0 0 0 0 0 0 1 0 0

0 0 0 1 0 0 0 0 1

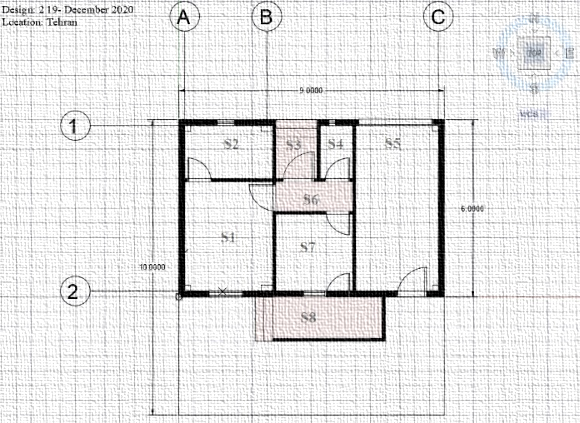
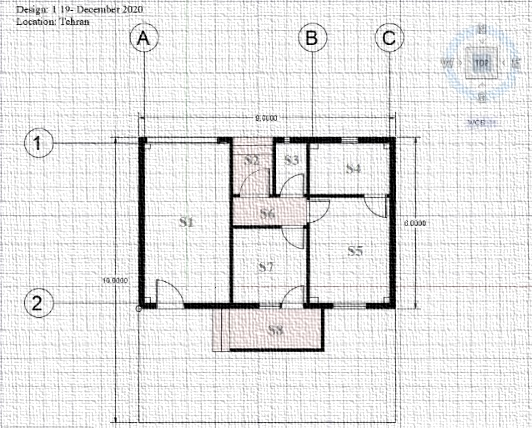
0 0 0 1 0 0 0 0 0

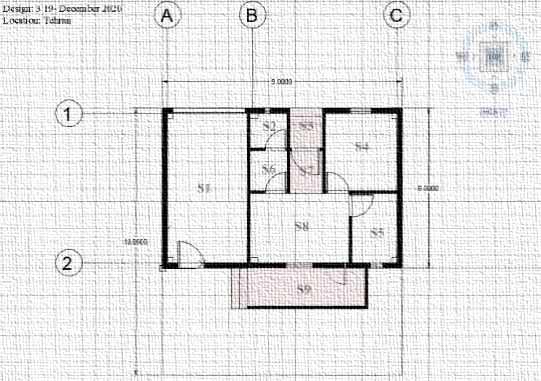
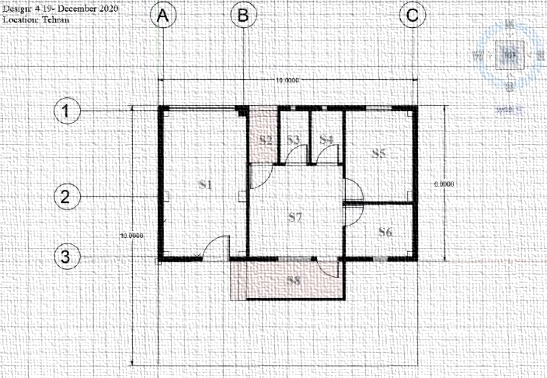
0 0 0 0 0 0 0 1 0

0 0 0 1 0 0 0 1 0

**Figure 10**. A matrix of spatial relationships and its transformation into a plan by automated design intelligence.

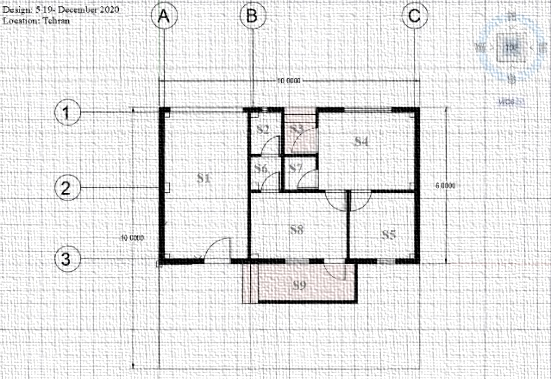
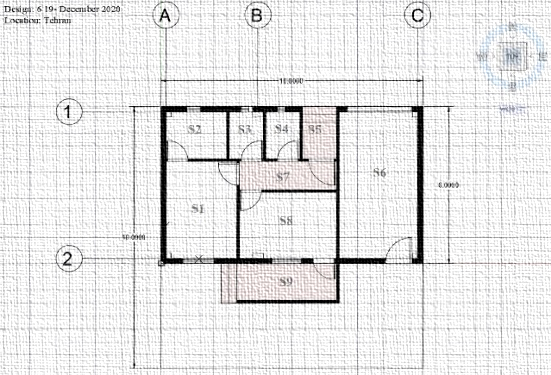
Notably, this method is used for columns and positioning doors and windows and their orientation, and after the matrices are prepared, it is written in Python programming language as special codes in combination with the library of artificial intelligence. Therefore, it can draw its automatic design intelligence as in Figures 11 to 16 in AutoCAD software.



**Figure 12**. Plan example drawn by automated design intelligence. **Figure 11**. Sample plan drawn by automated design intelligence

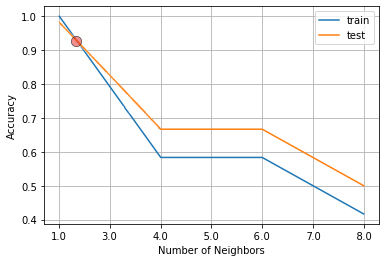
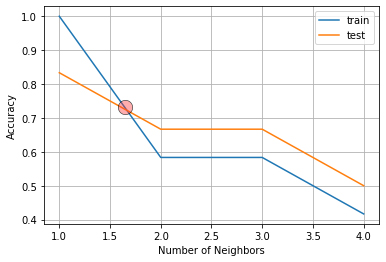
**Figure 14**. Plan example drawn by automated design intelligence. **Figure 13**. Sample plan drawn by automated design intelligence.

This is an example of plans that automatic design intelligence has calculated, designed, and drawn the occupancy level after entering data based on user interest and applying designing criteria and national building standards in a typical floor with spaces such as bedroom, kitchen, living room, parking, bathroom and toilet for a 10×20 land.



**Figure 16**. Plan drawing by automated design intelligence. **Figure 15**. Plan example drawn by automated design intelligence.

But to measure the learning accuracy of the algorithm and validation of the present study, the cyclin library and the learning and testing module have been used. In this way, 70% of the data in the database (dimension and size vectors of residential spaces) have been separated as educational data and trained to the machine and then 30% of the remaining data has been taken as test data after learning the algorithm from the test machine, this training and test has been repeated twice, the accuracy of which has been increased from 70% to 90% according to Figures 17 and 18.



**Figure 18**. Algorithm learning rate with 93% accuracy. **Figure 17**. Algorithm learning rate with 75% accuracy.

Figure 19 shows a diagram of the design process with algorithms (machine learning and evolutionary). In previous methods, an evolutionary or deep learning algorithm was used to generate the plan. In none of these researches was it possible to enter information and accompany the user for design, this is the first time that it is possible for the user to express their demands in the process of producing a plan with artificial intelligence.

In this research, the user first enters the features needed for the architectural plan in his user panel and the server then receives the information from the site and sends it as a vector of numerical properties to the core of the algorithm (k-means clustering). The algorithm then automatically opens AutoCAD software and creates a new page to start designing.

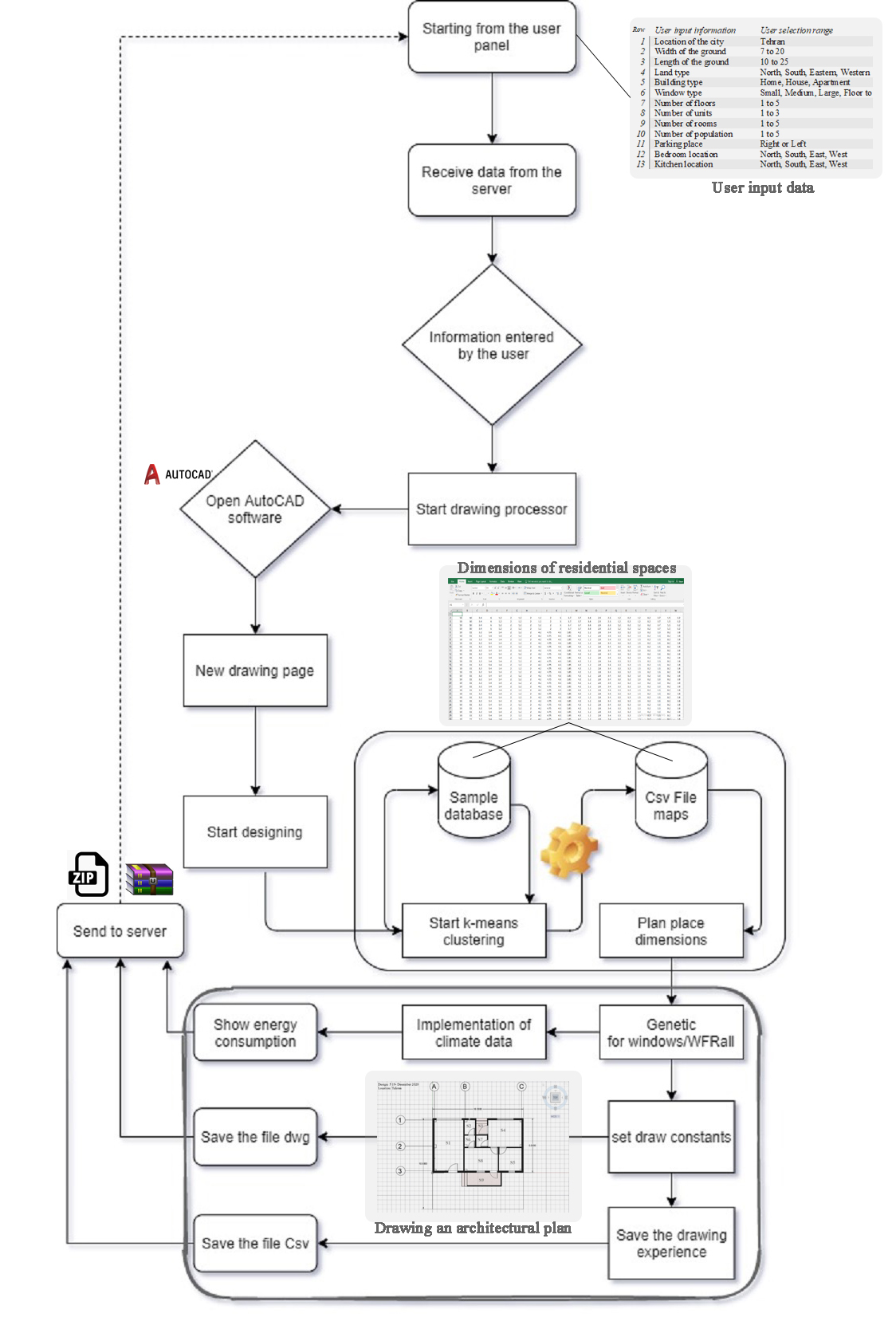
Then, the vector machine learning algorithm receives the features entered by the user from the server and then compares this vector with the feature vector of trained architectural plans with the help of Euclidean distance calculations and, selects the nearest feature from the csv database and labels it from 0 to infinity from smallest to largest, respectively.

The purposeful label, which represents the appropriate plan (based on the dimensions of the land and the user input features) with the trained samples, is now measured using the K-NN, k-nearest neighbor algorithm and, then the vector of the desired features enters the plan drawing process, in which Python code has been created with the help of special libraries (for this research), begins the process of plotting the plan in AutoCAD software.

Simultaneously with drawing the plans, the genetic algorithm equates the dimensions of the spaces received from the database with the dimensions of the user's land so that the dimensions of the spaces are drawn correctly. Also, the genetic algorithm with a minimal search structure seeks to optimize the dimensions of the various openings of the plan based on the computational methods of Ashrae and Delft of the Netherlands, which will deal with its working method separately in another study.

At the end of the processing procedure, after the final drawing of the map of plans and energy calculations and other processes, the file is transferred to the server in AutoCAD format as well as other related formats in the form of a zip file and then sent through the server to the user account requesting the architectural plan, also, the experience of this plot is stored as a CSV file with the dimensions and sizes, space syntax, room coordinates and other related calculations separately in the server, so that in the future, with the development of this artificial intelligence, stored experiences to increase diversity and quality of the architectural plans are evaluated and used.

But the limitations of the research, the limitation of this research in the initial step is the need for very strong and advanced equipment to implement servers and implement various simultaneous algorithms. Also, because such research is on the edge of modern knowledge and technology, the time and cost limitations of the research are another matter. The next limitation in the field of the operation of this algorithm is for other uses such as commercial, office, or medical, because this artificial intelligence has been trained only in the field of designing plans for residential use of villa type, and for other uses, related data should be collected and the necessary training should be given to artificial intelligence.

****

**Figure 19**. Part of the process of producing a plan with artificial intelligence.

**5. Conclusion**

The results of this study show that the use of feature vectors as supervised learning with k-nearest neighbor and k-clustering algorithms can lead to the identification of the best design and provide supervised relative awareness in machines, and the use of genetic algorithms to search the design space and measure the plans based on the dimensions of the designable land is a good solution, another point is that the time required for the plan design process by the algorithm will be in the range of 50 to 100 seconds if only one demand is sent from the server, which greatly shortens the time to reach the appropriate plan.

Another part of the research shows that with the help of Python libraries, the experience of the plan design process by the machine can be saved as CSV files. Finally, these design and plotting experiences are suggested to be used in reinforcement learning algorithms to produce more examples.

**Reference**

[1] N. Negroponte, “Through a Humanism Through Machines,” AD Architectural Design, volume 9(1969), doi/abs/10.5555/3289787.3289794.

[2]D. Nagy, D. Lau, J. Locke, J. Stoddart, Project Discover: An application of generative design for architectural space planning,(2017).

[3] R.T. Marler and J.S. Arora, “Survey of multi-objective optimization methods for engineering,” Structural Multidisciplinary Optimation 26 (2004), 369–395 <https://doi.org/10.1007/s00158-003-0368-6>.

[4] T. Murata, and H. Ishibuchi, , “MOGA: multi-objective genetic algorithms,” in Evolutionary Computation, IEEE International Conference on (1995), Vol. 1, p. 289.

[5] B. Hillier, et al. "Space syntax." Environment and Planning B: Planning and Design 3.2 (1976): 147-185.

[6] J. Peponis, J. Wineman, S. Bafna, M. Rashid, S. H. Kim, “On the generation of linear representations of spatial configuration,” Environment and Planning B: Planning and Design, 25 (1998), 559-576.

[7] A. Turner, M. Doxa, D. O’Sullivan, A. Penn, “From isovists to visibility graphs: a methodology for the analysis of architectural space”, Environment and Planning B: Planning and Design, 28 (2001), 103-121 <http://dx.doi.org/10.1068/b2684>.

[8] G. Renner, A. Ekrárt, Genetic algorithms in computer aided design. Computer-Aided Design (2003) ,35: 709-726,https://doi.org/10.1007/978-1-4020-8728-8\_21.

[9] DE. Goldberg, G. Rzevski, Genetic algorithms as a computational theory of conceptual design. Applications of Artificial Intelligence in Engineering(1991), VI: 3-16 <https://doi.org/10.1007/978-94-011-3648-8_1>.

[10] MA. Rosenman,An exploration into evolutionary models for nonroutine design. Artificial Intelligence in Engineering (1997),11: 287-293 <https://doi.org/10.1016/S0954-1810(96)00046-5>.

[11] MA. Rosenman, The generation of form using an evolutionary approach. In Evolutionary Algorithms in Engineering Applications. (1997), Berlin Heidelberg: Springer-Verlag, <https://doi.org/10.1007/978-3-662-03423-1_4>.

[12] JS. Gero, T. Schnier, Evolving representation of design cases and their use in creative design. In Proc 3rd Int Conf Comput Models Creative Design(1995).

[13] J. Poon, ML. Maher,Co-evolution and emergence in design. Artificial Intelligence in Engineering(1997), 11: 319-327, <https://doi.org/10.1016/S0954-1810(96)00047-7>.

[14] G. De Silva AG, Maher ML, Characterising evolutionary design case adaption. In Artificial Intelligence in Design. Dordrecht: Kluwer Acad(2000) <https://doi.org/10.1007/978-1-4020-8728-8_21>.

[15] P. Wilson, Computer supported cooperative work: an introduction. Kluwer Academic, Dordrecht(1991).

[16] C. Peng, Design through digital interaction: Computing communications and collaboration on design. Intellect Books(2001).

[17] A. Banerjee, Juan C. Quiroz and Sushil J. Louis A Model of Creative Design Using Collaborative Interactive Genetic Algorithms University of Nevada, Reno, USA(2008) <http://dx.doi.org/10.1007/978-1-4020-8728-8_21>.

[18] B. Preas, W. VanCleemput: Placement algorithms for arbitrarily shaped blocks. In: Proc. Design Automation Conf., pp. 474{480 (1979) ,https://doi.org/10.1145/62882.62904.

[19] S. Schneider, J. Fischer, R. K¨onig: Rethinking automated layout design: Developing a creative evolutionary design method for the layout problems in architecture and urban design. DCC pp. 367{386 (2011).

[20] K.Knecht, R. Koenig: Generating floor plan layouts with k-d trees and evolutionary algorithms. In: Generative Art Conf., pp. 238{253 (2010).

[21] P.Merrell, E. Schkufza, V.Koltun: Computergenerated residential building layouts. p. 181. ACM (2010), <https://doi.org/10.1145/1882261.1866203>.

[22] M. Harada, A. Witkin, D. Baraff: Interactive physically-based manipulation of discrete/continuous models. In: Proc. CGIT, pp. 199{208 (1995).

[23] N. Umetani, T. Igarashi, N.J. Mitra: Guided exploration of physically valid shapes for furniture design. ACM Trans. on Graphics 31(4), 86:1{86:11 (2012).

[24] A.R. Ahmad, O. Basir, K. Hassanein and M.H. Imam. A hierarchical placement strategy for generating superior layout decision alternatives.International Journal of Operations and Quantitative Management,(2005).11, pp. 261-280.

[25] J. Brotchie, M. Linzey, A model for integrated building design, Building Science 6 (3) (1971) 89–96, <http://dx.doi.org/10.1016/0007-3628(71)90020-X,(URL>

[26] R.S. Liggett, W.J. Mitchell, Optimal space planning in practice, ComputerAided Design 13 (5) (1981) 277–288, <http://dx.doi.org/10.1016/0010-4485(81)> 90317-1.

[27] K.Y. Lee, M.i Roh, H.S. Jeong, An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages, Computers and Operations Research 32 (4) (2005) 879–899, <http://dx.doi.org/10.1016/j.cor.2003.09.004>.

[28] T. Irohara, M. Goetschalckx, Decomposition solution algorithms for the multi-floor facility layout problem with elevators, 19th International Conference on Production Research (ICPR), Valparaiso, Chile, July 29-August 3, (2007).

[29] H.H. Huang, M.d May, H.M. Huang, Y.w Huang, Multiple-floor facilities layout design, Proceedings of 2010 IEEE International Conference on Service Operations and Logistics, and Informatics, IEEE, ISBN: 978-1-4244-7118-8, (2010), pp. 165–170, <http://dx.doi.org/10.1109/SOLI.2010.5551588>

[30] R.S. Liggett, Automated facilities layout: past, present and future, Automation in Construction 9 (2) (2000) 197–215, <http://dx.doi.org/10.1016/S0926-5805(99)> 00005-9

[31] A. Doulgerakis, Genetic Programming + Unfolding Embryology in Automated Layout Planning. (Msc thesis) Bartlett School of Graduate Studies, University College London, (2007) <https://discovery.ucl.ac.uk/id/eprint/4981>.

[32] R.W.J. Flack, Evolution of Architectural Floor Plans, Tech. Rep, Brock University, (2011) <https://doi.org/10.1007/978-3-642-20520-0_32>.

[33] G. Zimmermann, From floor plan drafting to building simulation — an efficient software supported process, in: I. Beausoleil-Morrison, M. Bernier (Eds.), International IBPSA Conference Building Simulation. Montreal, Quebec, Canada, (2005), pp. 1441–1448.

[34] D. Crasto,A. Kale,and C. Jaynes, "The Smart Bookshelf: A study of camera projector scene augmentation of an everyday environment," Proc. 7th IEEE Workshop on Applications of Computer Vision, WACV'05,Jan (2005),pp I - 8 <http://dx.doi.org/10.1109/ACVMOT.2005.116>.

[35] J. Y. Lee,M. S. Kim,and J. J. Lee, "Design of Fuzzy Controller for Car Parking Problem Using Evolutionary Multi-objective Optimization Approach," Proc. IEEE International Symposium on Industrial Electronics,July 9- 12,(2006),pp 329-334 <http://dx.doi.org/10.1109/ISIE.2006.295615>.

[36] M. Keckeisen, M. Feurer, and M. Wacker, "Tailor Tools for Interactive Design of Clothing in Virtual Environments," Proc. ACM Symposium on Virtual Reality Software and Technology,Nov 10-12,(2004),pp 182-185.

[37] J. J. Michaleka, R. Choudhary, and P. Y. Papalambrosa, "Architectural Layout Design Optimization," Eng. Opt., (2002), Vol. 34(5),pp. 46 1—484, DOI: 10.1080=0305215021000033735.

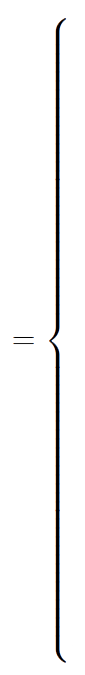
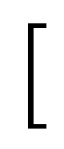
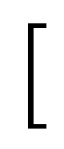
[38] P. H. Levin, "Use of graphs to decide the optimum layout of buildings," Architect-(I964), 14,pp 809-8 15

[39] M. Verma and M. K. Thakur, "Architectural Space Planning using Genetic Algorithm," Proc. 2nd International Conference on Computer and Automation Engineering, ICCAE(2010), Feb 26-28, Vol 2, pp 268-275, <https://doi.org/10.1109/ICCAE.2010.5451497>.

[40] M.K. Thakur. M. Kumari, Architectural Layout Planning using Genetic Algorithms, 978-1-4244-5540-9/10©(2010) IEEE, <https://doi.org/10.1109/ICCSIT.2010.5565165>.

[41] E. Rodrigues a,⁎, A. Rodrigues Gaspar a, Á. Gomes. An approach to the multi-level space allocation problem in architecture using a hybrid evolutionary technique. 2013 Elsevier B.V. All rights reserved.http://dx.doi.org/10.1016/j.autcon.2013.06.005.

A.1.Mathematical model

 (A.1)

0

1

Tehran

LC =

HOM

HOU

AP

BT =

N,S,E,W

LT =

N

S

E

W

0

1

2

3

(A.2)



MLGAFPG

HOM, HOU, AP

0

1

2

(A.3)

WG =

 (A.4)

7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 (A.1)

7 to 20



10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25

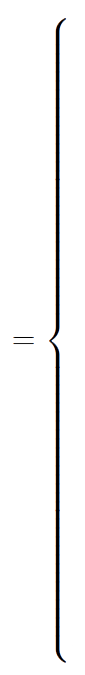
LG =

(A.5)

10 to 25

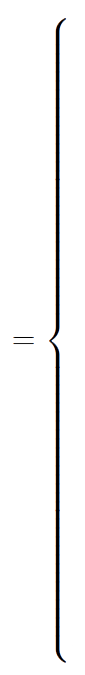
NF =

NU =



1,2,3,4,5

1 to 5

 (A.11)

0

1

KL =

N,S,E,W

N,S,E,W

BL =

PP =

Ri Le

Ri

Le

 (A.6)

1,2,3

1 to 3

 (A.12)

0

1

2

3

N

S

E

W

 (A.7) (A.13)

1,2,3,4,5

1 to 5

NR =



(A.14)

0

1

2

3

1,2,3,4,5

1 to 5

NP =

N

S

E

W

 (A.8)

MLGAFPG

MLGAFPG

0

1

2

SM

ME

LA



0

1

CLO OPE

SM, ME, LA

WT =

(A.9) (A.15)

CLO ,OPE

KM =



0

1

2

11111100

BMAS

BTO

BAT



0

1

Yes

No

Yes, No

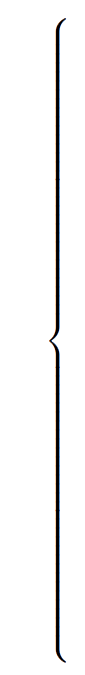
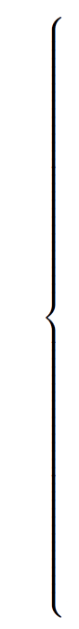
COR =

BMAS, BTO, BAT

BM =

(A.10) (A.16)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Table.Nomenclature |  |  |  |  |
|  | (A.1) | User selection range |  | (A.9) | Number of units |
|  | (A.2) | Location of the city |  | (A.10) | Number of rooms |
|  | (A.3) | Width of the ground |  | (A.11) | Number of population |
|  | (A.4) | Length of the ground |  | (A.12) | Parking place |
|  | (A.5) | Land type |  | (A.13) | Bedroom location |
|  | (A.6) | Building type |  | (A.14) | Kitchen location |
|  | (A.7) | Window type |  | (A.15) | Kitchen model |
|  | (A.8) | Number of floors |  | (A.16) | Bathroom model |



0 (A.1)

1 (A.2)

0 (A.3)

1 (A.4)

0 (A.5)

0 (A.6)

1 (A.7)

0 (A.8)

0 (A.9)

2 (A.10)

1 (A.11)

1 (A.12)

1 (A.13)

2 (A.14)

2 (A.15)

1 (A.16)



UIIV

UIIV =

0, 1, 0, 1, 0, 2, 0, 0, 1, 0, 0, 2, 1, 1, 1, 2, 2, 1



0, 1, 0, 1, 0, 2, 0, 0, 1, 0, 0, 2, 1, 1, 1, 2, 2, 1

D IIt = (Vector plan information in the database)



0, 1, 0, 1, 0, 2, 0, 0, 1, 0, 0, 2, 1, 1, 1, 2, 2, 1

U IIV = (Vector plan information from user input data)

=

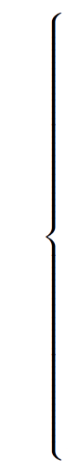
D IIt (0, 1, 0, 1, 0, 2, 0, 0, 1, 0, 0, 2, 1, 1, 1, 2, 2, 1) and U IIV (0, 1, 0, 1, 0, 2, 0, 0, 1, 0, 0, 2, 1, 1, 1, 2, 2, 1).

d(u⃗ ,v⃗ ) = ∥u⃗ − v⃗ ∥

d(u⃗ ,v⃗ ) = ∥u⃗ − v⃗ ∥ =

d(u⃗ ,v⃗ ) = ∥u⃗ − v⃗ ∥ =

d(u⃗ ,v⃗ ) = ∥u⃗ − v⃗ ∥ = 0 (No differences in input data and learning algorithm data)

 U1 = (0X1,4 Y1),(0 X3,11 Y3),(5 X4,4 Y4),(5 X2,11 Y2)

A.2.Connectivity/Adjacency

U2 = (5X1,7 Y1),(5 X3,11 Y3),(7X4,7 Y4),(7 X2,11 Y2)

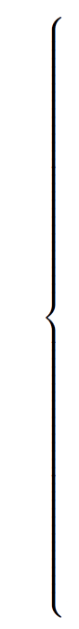
U3 = (7X1,7 Y1),(7 X3,11 Y3),(9X4,7 Y4),(9 X2,11 Y2

Machine learning genetic floor plan generation (MLGAFPG) U4 = (9X1,7 Y1),(9 X3,11 Y3),(12X4,7 Y4),(12 X2,11 Y2)

U5 = (12X1,0 Y1),(12 X3,11 Y3),(18X4,0 Y4),(18 X2,11 Y2)

U6 = (5X1,0 Y1),(5 X3,7 Y3),(12X4,0 Y4),(12 X2,7 Y2)

U7 = (0X1,0 Y1),(0 X3,4 Y3),(5X4,0 Y4),(5 X2,4 Y2)



md1

md2

X2 = A2 + B2

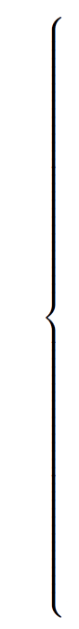
U1 = (0X1,4 Y1),(0 X3,11 Y3),(5 X4,4 Y4),(5 X2,11 Y2) X2 = 72 + 52

X2 = 74

X = = 8.60

md3

md4



md1

md2

X2 = A2 + B2

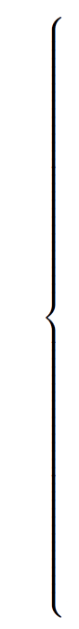
U2 = (5X1,7 Y1),(5 X3,11 Y3),(7X4,7 Y4),(7 X2,11 Y2) X2 = 22 + 42

X2 = 20

X = = 4.47

md3

md4

 md1

md2

X2 = A2 + B2

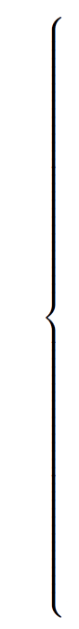
U3 = (7X1,7 Y1),(7 X3,11 Y3),(9X4,7 Y4),(9 X2,11 Y2) X2 = 22 + 42

X2 = 20

X = = 4.47

md3

md4



md1

md2

X2 = A2 + B2

U4 = (9X1,7 Y1),(9 X3,11 Y3),(12X4,7 Y4),(12 X2,11 Y2) X2 = 32 + 42

X2 = 25

X = = 5

md3

md4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Table.Nomenclature |  |  |  |  |
|  | **S** | Space |  | **Fh** | Floor height |
|  | **E** | Elevator |  | **Fr** | boundary of spaces |
|  | **W** | Width of the ground |  | **Wr** | Windows |
|  | **L** | Length of the ground |  | **Dr** | Doors |
|  | **ST** | Stairs |  | **Fri** | Degrees of freedom |
|  | **Le** | Level |  | **OLA** | Final area |
|  | **P** | Parking |  | **Ol** | Occupancy level |
|  | **W** | Warehouse |  | **fa** | Area of spaces |
|  |  | Connecting matrices of spaces |  |  | Neighborhood matrix |
|  |  | Communication space matrix |  |  |  |