A Mass Customization Oriented Housing Design Model Based on Genetic Algorithm

Özge Güngör¹, Gülen Çağdaş², Özgün Balaban³.
¹Istanbul Technical University, Architectural Design Computing, Turkey, ²Istanbul Technical University, Architecture, Turkey, ³Istanbul Technical University, Architectural Design Computing, Turkey.
¹ozgenexa@yahoo.com, ²glcagdas@gmail.com, ³ozgunbalaban@gmail.com.

Abstract. Today, right along with the products marketed and manufactured by the mass production techniques, continuously developing computing and technology have an undeniable impact on customized design, in which the users have a say on the products design and manufacturing. Mass customization is slowly settling down in architectural design concepts as well, like housing which is one of the best areas where users can reflect their living habits and preferences. In this study, user centric mass customization based model is developed, which creates housing floor plan combining the user-supplied data with the best possible creations generated by the genetic algorithms.

Keywords. Architectural Design Computing; Housing Design; Genetic Algorithm; Mass Customization.

INTRODUCTION

Housing is where people’s basic needs are met. Within everyone’s house, there lies the most private areas of its users and these users are attached to their places. Individuals’ expectations from their houses are by satisfying their personal needs and wishes, increasing or at least protecting their living standards. Therefore, people are instinctively prone to design and build their own houses. Since the beginning of the twentieth century until now, residential production is done with the help of industrial products and tools. In this rapidly industrializing of construction era, user participation in design process is neglected. This result in identical buildings that are built with the logic of mass production and using industrialized products, and these housings’ failure to fully satisfy every individual residing in these houses is an important problem. Residential design that is organized according to the needs of the user, is inevitable to satisfy users.

Mass customization is designed to deliver highly customized products with mass production efficiency (Piller and Kumar, 2006). Mass customization plays a very important role in the usability of the products as in different levels of the work like design, manufacturing and distribution it is always in contact with the users. It is in fact standing out in many fields like...
computer hardware and software, particular shoes and outfit, and in the automotive industry. Pine (1993) indicates that advanced technology applications, computer integrated manufacturing systems, computer-aided design and advanced computer technologies are the prerequisites for mass customization to work correctly. With the software and hardware of support of computer-aided design systems, mass customization began to be used in construction. There are not many mass customized design examples until now, and the rare applications are generally catalog software which the user selects the materials. However one past study done long before computers participated in design practice, become starting point of this work. It is a semi customization project that is done by “United Architects” in İzmit between 1971 to 1977. In this project users who are factory workers with low income participated in design process and as a result houses that are fulfilling the needs of its users are constructed (Çavdar 1978). In addition, the project was both economically reasonable and constructed fast.

A computer aided design tool, which produces floor plan and building form by processing user supplied information within a genetic algorithm, is developed to design mass customized residential buildings. Genetic algorithms or as a more broad sense evolutionary algorithms are computational problem solving techniques that use evolutionary processes already existing in nature. Problems that are difficult to solve with traditional programming techniques, especially classification and multidimensional optimization problems are solved easier and faster with the help of genetic algorithms (Atalağ 2001).

The main reason of using genetic algorithms for creating models is that genetic algorithms may lead to much more diverse solutions to a problem than traditional design process. In genetic algorithms, evolutionary processes void most of the inadequate individuals, thus there is always improvement in designs. Nonetheless, not every unfit individual is killed; some of them are saved to help diversity in designs. Genetic algorithms’ ability of constant improvement along with diversity helps users find variety of choices suiting their needs.

In this model, the aim is not giving all reins to the user, as ordinary user might not be fully informed about his needs. But instead this model tries to develop a channel of information between user and the architect. In this attempt we are using genetic algorithms to show various different design solutions to both the architect and the user who may not realize all the options by traditional design approaches.

**MODEL FOR HOUSING DESIGN BASED ON GENETIC ALGORITHM**

The primary aim of the model developed in this study, is to involve user participation within the designing process. To succeed in creating a mass customization model, one must use architectural design computing combined with user friendliness. To make the program open to non-expert user participation, the model starts from the most basic component i.e. room design and it is constantly evolved to form full model of a housing element. Therefore, the model uses bottom-up design approach. As for computational component of the model, in determining the geometries of the rooms, routines of the genetic programming are applied thus it is possible to generate plenty of possible room designs, which is nearly impossible in traditional design approach. By these features, the model suits mass production coupled with user customization and in turn they produce mass customization model. The hierarchy within the program starts from the smallest components genes, genes get together to build rooms and rooms are combined to build floors and floors finally make up housing elements. Matlab Scientific Development Suite is used in the implementation of the model.

With the start of the program, parameters of the genetic algorithm are determined. These are population size, elitism factor, crossing and mutation ratio, which are determined after series of experiments. These parameters are embedded within the algorithm and the user cannot change these values.

When using genetic algorithm for generating room components of a building, genes and
chromosomes that are the fundamental components of the algorithm must be defined. Gene structure used in this model must be defined for the whole space that the model uses and genes must be unique meaning no two other gene carries the same information. Furthermore, gene structures should not change and it should be preserved within all chromosomes, for concepts like gene transfers within chromosomes and genes getting mutations work.

Each one of the genes that are representing one unit square component, i'th gene can be defined as X and Y coordinates which in turn indicates the location of the gene.

\[ \text{Gene}(i) = \{ \text{Component}(i).\text{Coordinate}_X, \text{Component}(i).\text{Coordinate}_Y \}, i=1,2,\ldots,N \] (1)

Since the chromosomes represent the genotypes of the rooms, a room that is N unit square, consists of N different genes. So, a chromosome belonging to the j'th room that has N unit squares, is defined like this:

\[ \text{Chromosome}(j) = \{ \text{Gene}(1);\text{Gene}(2);\ldots;\text{Gene}(k) \} = \{ \text{Component}(i).\text{Coordinate}_X,\text{Component}(i).\text{Coordinate}_Y \}, i=1,2,\ldots,N \] (2)

For a chromosome to represent valid room geometry, it must satisfy size condition as well as, the components must not intersect or there should be no spaces between two components. Thus, a chromosome cannot be structured randomly from the genes. To ensure valid chromosome, chromosomes are constructed by adding new genes randomly only to the neighboring spaces. First step in creating a chromosome is to take the first gene as \{0,0\} gene (gene residing in the 0,0 coordinate). From the four neighbors of this gene, one of them is randomly selected and is added to the gene set. After that, the new gene is selected from the neighbors and this carries on until the chromosome has the desired size. The chromosome generating algorithm repeats itself until population size is reached.

The initial chromosomes generated are evaluated by a fitness function. Fitness function should be a mathematical function. For example, in room generation circumference/area ratio can be used. So, the fitness function becomes:

\[ \text{Fitness Function} = \min(\sum(\text{ComponentSideNumber})/\sum(\text{ComponentAreaSize})) \] (3)

The chromosomes are sorted according to this fitness function; the ones that have less circumference/area ratio meaning their shapes are more straight than others, will have higher ranks. However, in this model, the aim is not to optimize the solutions thus the chromosomes that are lower in rank is not disregarded; on the contrary, they are preserved to add diverse solutions. After the sort, some chromosomes that are in the highest ranks are separated by elitism function. The aim in the elitism function is to avoid mutations and crossings for the most successful chromosomes. The remaining chromosomes are crossed over each other randomly. In the crossing over, the critical step is to control each gene to ensure that there is no space between the genes, if this condition is not satisfied than the new breed is disposed. (Figure 1) In the end, all the solutions are gathered and this time they are evaluated against the...
criteria that users entered. These criteria can be facing to scenery, such that the biggest side of the room looks to a certain direction. If there is no other condition, the generated rooms are stored in the database.

Floor plan generating algorithm starts by retrieving the generated room shapes from the database. Then all the rooms situated in one floor are drawn one by one within the borders of that floor and every room is checked to ensure no room is intersecting another room. If there is any intersecting room, then this plan is cancelled, and to find a new suitable plan every room is shifted in the x and y axis to search for no overlapping plans. This search continues until a suitable plan is found.

THE INTERFACE OF THE MODEL
This model, which aims mass customization in architectural design process, needs to combine the genetic algorithm with user interaction. To succeed in this, a self-explaining user interface that can even make possible for a novice user to use the program, is designed. There are three main screens in the model; data entry interface, room geometry interface and floor plan interface.

Data entry interface is the main screen, which greets the user and gets information from him. In the first data interface, the user enters certain properties of the site where the building will be erected and residential type (apartment block, single detachment etc...) of this building. If there are some restrictions about the site dimensions user selects restricted option and enters the dimensions to the site properties panel, else he selects unrestricted and only enters the required area. Afterwards if there is a dominant direction where there is a road, scenery etc..., that direction

![Figure 2](https://example.com/image2.png)

**Figure 2** Interface for entering topological features of the rooms
is marked. Within the ‘Residential type’ panel, the user enters number of floors if it is a detached building, else if it is an apartment block he enters floor number.

The user enters information about the rooms that are intended to be in the building in the second data interface. In this screen various room types are listed such as kitchen, bedroom, study room etc… and if the user wants a certain kind of room, he enters desired area of the room, if he does not want that room, leaves it blank. As the user selects rooms, he can see remaining area, from the occupancy panel that is at the bottom of the screen. If desired some rooms can be fixed to any direction. After that, if the user wants to have bathroom adjacent to bedroom, or dining room with living room, he can make it so in this screen. Furthermore, if the user has more than one floor to design this step is repeated. The resulting information is saved into the database and the program moves on to the choice selection interface.

The shapes of best results of all rooms, which are created by genetic algorithms and selected with the fitness function, are shown via the display panel of the first choice selection interface. The user can determine performance scores of each individual that is created by evolution within the population and from the database panel he can select whichever individual he wants. If later he likes to remove some room forms, this can be done in the same panel. After the user finishes selecting room shapes for all the desired rooms, he will continue from the next selection interface.

One other selection interface is floor plan selection interface, which allows the creation of floor plans as seen in figure 3. In this panel rooms that are displayed with their code numbers, Here, on the
main panel the rooms that are joined together according to the typological restrictions given by the user, can be selected. 'Detailed view' button that is found on the database panel, will enlarge the floor plan and visualize it with a two dimensional image. Like the other selection interfaces, the user can see how well his floor plan is depending on the fitness function from the performance graphic. Because the choices are left to the users instead of automatically selecting the best designs, the model complies with the logic of mass customization.

To display the created floor plans and form of the buildings, graphical interfaces are used. At this interface, selected floor plans can be observed as two-dimensional diagrams. If the user wants to examine one of the floor plans in more detail, he chooses it via the operator panel.

The user can pan, zoom or rotate the three dimensional model of selected floor plan using the mouse, to get a better understanding of the model (Figure 4). In the three-dimensional graphical interface, rooms are represented with different colors, and there is a legend showing which color is representing which room. In the last stage, the user selects the most suitable floor plan after experiencing it as three-dimensional model, and stores in a database to send it to the architect.

By using genetic algorithm for mass customization of housing design model, one can design different types of housing units that will be installed on different terrains and suitable for number of different people. One of the different types of housing is one that has specific land borders, is 3-storey and is designed by a family with many children can be given as an example. As shown in figure 5 a detached

![Figure 4: Graphical interface that allows three-dimensional visualization](image)
house that has specific boundaries of land, has bathrooms associated with the bedrooms and has a living space in each floor is designed. Every square in the field, corresponds to a square meter. Vertical circulation in the building, is placed in the same place for each floor. After the algorithm finishes design process void spaces in the building mass can be used as balcony, terrace, wardrobes, air conditioning and plumbing units.

In another possible scenario, each user is known in which floor of a multi-storey apartment they will be residing, after each of the users designed their homes and saved into a database the resulting housing can have a form that is similar as in figure 6. After leaving fixed spaces like entry and vertical circulation of the building, user can change their own floor plans according to their tastes and send it to the architect to discuss on it.

All the information regarding a floor plan are saved into the database, as a result users can take their neighbors’ floor plan if they like it and apply it in their own without any changes or with some small modifications. In addition, buildings designed with this model can be built with prefabricated building elements thus building process will be shorter and costs will be lower.

CONCLUSION
In this study, housing design model, which imitates natural evolutionary process of the genetic algorithms and combines users’ preferences and allows them to customize their housing, is developed. This model is thought to be used by users who want to design their own private housing by participating in the design process or different people using same apartment block but customizing their houses according to their needs and finally showing what they have in mind to the architect. Besides, the model can be developed in the future so that it can use different angles and curves to allow designs that are more flexible. Furthermore as a future goal, the ability to export the models into two or three dimensional architectural design tools will be added. By this way architects can quickly work on the models and develop them further.

REFERENCES