Floor Layout Planning Using Artificial Intelligence Technique

Mitali Chavan, *Student, PCE*, Nidhi Menon, *Student, PCE*, Ritika Kumar, *Student, PCE*,
Shivani , *Student, PCE*, and , Prof. Varunakshi Bhojane, *PCE*

***Abstract*—In the era of e-commerce while buying furniture online the customers obviously feel the need for visual representation of the arrangement of their furniture. Even when doing interiors of the house it's difficult to just rely on assumptions about best layouts possible and professional help may become quite expensive. In this project, we make use of Genetic Algorithm (GA) which is an Artificial Intelligence technique to display various optimal arrangements of furniture. The basic idea behind using GA is developing an evolutionary design model. This is done by generating chromosomes for each possible solution and then performing a crossover between them in each generation until an optimum fitness function is reached. Modification in chromosome representation may also be done for better results. The proposed system will generate different layout designs for the furniture keeping in consideration the structure of a master bedroom.**

***Index Terms*—chromosome, genetic algorithm, genetic operators, layout generation**

I. Introduction

In architecture and building engineering, a floor plan is a drawing to scale, showing a view from above, of the relationships between rooms, spaces and other physical features at one level of a structure. Floor plans may include notes for construction to specify finishes, construction methods, or symbols for electrical items. The floorplanning can be classified into two categories either an equal area layout problem or an unequal area layout problem. The equal area layout problem is to determine how to assign a set of distinct facilities, to a set of distinct locations, so that each facility can be assigned to a single location called as one-to-one assignment problem. The unequal area layout problem is to regulate the allocation of all facilities within a block plan (or available area).

Optimizing furniture arrangement into a realistic and functional indoor configuration involves considerable complexity, taking into account various interacting factors, such as pair wise furniture relationships, spatial relationships with respect to the room, and other human factors. An effective representation that captures the necessary spatial relationships is needed. Some of the attributes to be considered while deciding furniture placement are - Bounding surfaces, Centre and orientation, Accessible space, Viewing frustum, Hierarchical relationships between different furniture objects.

II. Scope

The proposed system will generate multiple layout designs for the furniture keeping inconsideration the structure of a master bedroom. Its dimensions will be given as input by the user. The master bedroom will have a window and two doors. The size of the doors and the window will be predefined and not subject to any changes but their position would be given by the user. The furniture elements are Wardrobe, Bed and Table. The size of these elements will also be fixed. The input given will be taken from the user in text format via a GUI and the output will be multiple possible arrangements of the furniture elements represented in 2D format.

III. Existing System

The existing systems for floor layout planning work on the concept of drag and drop mechanism. The user is shown a layout of the room or is asked to choose a layout from the options provided which seem similar to the layout of the room they wish to plan and then from the set of objects provided they are expected to use the drag and drop facility. There is generally a set of symbols, realistic images or clip art images provided for the different kinds of furniture elements and the user is expected to only drag them on the room layout they find similar to their room and drop it on the location they wish to place it. The whole planning and layout produced can be in 2D or 3D format. Though this advantage of a 3D view of the room is given to the user it can also be seen that the existing system merely allows him to visualize how his thought-of room layout would look like or how he could think of a little alterations to his plan to result in a different layout. This system also has room for human errors of planning for instance, not leaving enough space for the door of the room to open and placing a table in front of it. Even an error of a centimeter is enough to make a layout inefficient.

Drawbacks:

· Though the user is able to see how exactly his layout will look like there are chances of him missing out more efficient layouts that the proposed system can provide.

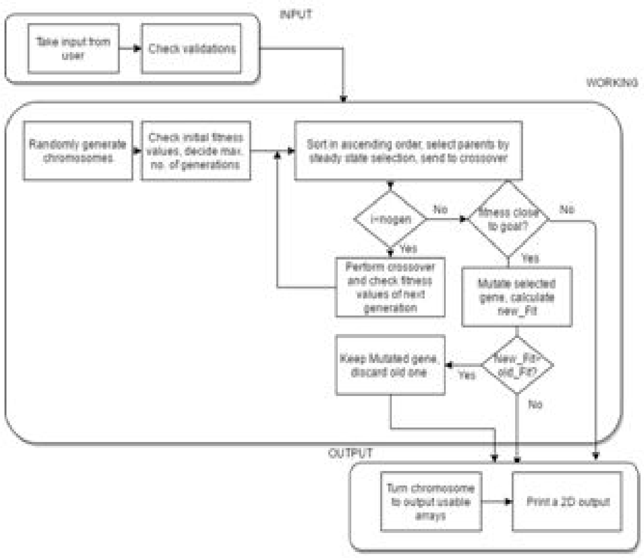
· A user might always not find a room layout similar to his room for selecting and placing the furniture elements and might have to compromise with the output.

- The furniture elements used may not be scalable to the room dimensions when placed and thus, may create difficulty for the user.

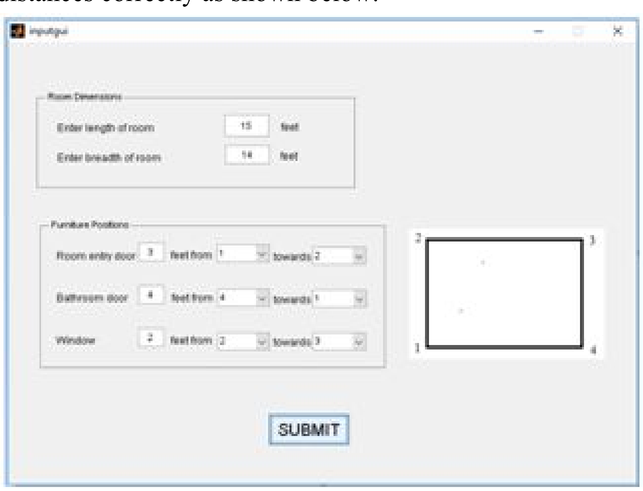
IV. PeraNoto

In Javanese "Furniture" is known as "Perabot" and "Arrangement" is known as "Noto". Thus combining the two, the proposed system is named "PeraNoto".

Use The proposed system follows all the stages of Genetic algorithm as shown in the flow chart below.



The input is taken from the user through the GUI where the user enters length, breadth and the distance of the doors and window from one point towards another point on a wall. The GUI has a rectangle displayed on the right which number the corner points to help the user measure the distances correctly as shown below.



The system then generates a randomized population of chromosomes. These chromosomes contain information about the coordinates of left bottom corner and also their alignment i.e. horizontal or vertical.

X_B	Y_B	A	X_T	Y_T	A	X_W	Y_W	A
1	2	3	4	5	6	7	8	9

The above figure gives representation of a single chromosome. Each furniture object has a designated sub-range of the array to code information specifically about it. The designated ranges are as follows

- sub- range (1:3) represents Bed
- sub-range (4:6) represents Table
- sub-range (7:9) represents Wardrobe

Each of the positions 3, 6 and 9 represent the alignment for the furniture represented by the respective sub-ranges to which these positions belong. These positions are set to 1 is alignment is horizontal and 2 if the alignment is vertical. The remaining two positions in each of the sub ranges give the coordinates of the bottom left corner of the furniture object that they represent. The coordinates are generated randomly and the maximum upper limit of these randomly generated numbers is the maximum value among length and breadth of the room whichever is greater. The alignment of these furniture elements; whether 1 or 2 is also random. The next step is to calculate fitness value of each of the chromosome in the population.

The fitness value of each chromosome is then calculated based on various constraints such as - the furniture objects do not overlap, they do not exceed the room dimensions, arrangement of furniture nearer to the walls is favoured, no furniture is in front of the two doors and wardrobe should never be in front of the window.

In addition to the score there are also flags included to notify certain conditions so that occurrences of false positives where a chromosome shows high fitness value score even after not fulfilling some basic conditions are identified. The scores assigned for each object were decided on the basis of trial and error method by experimenting with various random values and are subject to change as per the needs of the application.

The fitness scores assigned accordingly are stated below.

Constraints	Wardrobe				Bed				Table			
	Min Score	Max Score	Flags		Min Score	Max Score	Flags		Min Score	Max Score	Flags	
			Yes	No			Yes	No			Yes	No
Overlap with Bed	0	10	1	0	-	-	-	-	-	-	-	-
Overlap with Table	0	10	1	0	0	10	1	0	-	-	-	-
Within room dimensions	0	10	1	0	0	10	1	0	0	10	1	0
In front of room door	0	10	1	0	0	10	1	0	0	10	1	0
In front of washroom door	0	10	1	0	0	10	1	0	0	10	1	0
In front of window	0	10	1	0	5	10	-	-	7	10	-	-
Near to any walls	5	10	-	-	5	10	-	-	5	10	-	-
Total Max/Min Score	5	70			10	60			12	50		

As evident from the table individual fitness of each object is calculated based on certain parameters.

- Fitness score of Wardrobe is calculated out of 70
- Fitness score of Bed is calculated out of 60
- Fitness score of Table is calculated out of 50

Hence Maximum Fitness Score of a Chromosome = ∑ Maximum Fitness score of each Furniture object

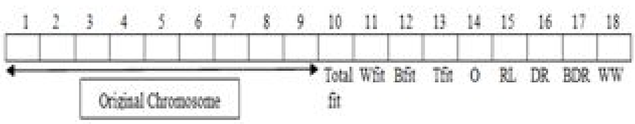
= Maximum score of (Wardrobe + Bed + Table)

= 70+60+50

= 180

Thus, Total Fitness score of a chromosome is always calculated out of 180.

Next we discuss flags; the flags give information about occurrence of conditions such as overlap, furniture exceeding room dimensions, furniture in front of any of the doors and wardrobe in front of the window. If a condition occurs then the flag is set to 1 otherwise set to 0. These conditions are checked for each object and then finally the flags for a chromosome are set by performing logical OR operation between the values of flags obtained for each object.



The above image represents the structure of the chromosome after its fitness has been calculated and related values are appended at the end of the chromosome. The flags are represented at positions 14,15,16,17 and 18.

Flag O: Flag is set to 1 if Overlap condition occurs

Flag RL: Flag is set to 1 if furniture objects exceeds room dimensions

Flag DR: Flag is set to 1 if furniture objects are in front of room entry door

Flag BDR: Flag is set to 1 if furniture objects are in front of bathroom door

Flag WW: Flag is set to 1 if Wardrobe is in front of window

To decide whether these flags should be set to 0 or 1 first the above flag conditions are checked for each furniture object and accordingly local flags are set to 1 or 0 then logical OR operation is performed on these local flags to set the chromosome flags O, RL, DR, BDR thus these flags will be set to 1 even if anyone furniture object breaches any constraint. The last flag WW only displays exclusive information about Wardrobe and hence does not include any other operations.

The positions 10, 11, 12 and 13 represent Total fitness score (Totalfit), Wardrobe fitness score (Wfit), Bed fitness score (Bfit) and Table fitness score (Tfit) respectively.

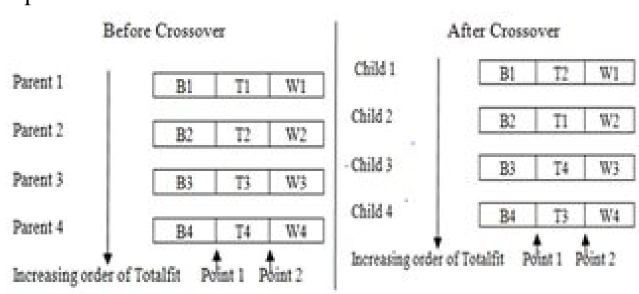
- Totalfit - Fitness value scored out of 180 for the entire chromosome
- Wfit - Fitness value scored out of 70 for the arrangement of Wardrobe
- Bfit- Fitness value scored out of 60 for the arrangement of Bed
- Tfit- Fitness value scored out of 50 for the arrangement of Table

Totalfit = ∑ Fitness value scored for each furniture object

= Wfit + Bfit + Tfit

The next step after evaluating fitness score is to select parent chromosomes for crossover. The selection operator used here is Steady State selection. This selection operator sets a minimum threshold fitness value and all the chromosomes with a higher fitness value than the threshold is allowed to participate in the crossover. For the system, not only do the chromosomes have to possess a Totalfit value higher than the threshold but also all the above mentioned flags should be set to 0 for them to qualify for the crossover. This additional measure helps to be assured that only good chromosomes are passed on to create next generation.

The crossover performed here is Two-Point crossover. The chromosomes are sorted in ascending order according to their Tfit value and then crossover is performed between a chromosome and its immediately next chromosome. Since, crossover should be performed only within 3 furniture objects and the internal coordinates or alignment of each object should not be disturbed sub-ranges for Bed(B), Table(T), and Wardrobe(W) are each treated as separate blocks



As seen in the above images, Two-Point crossover is done by replacing only the Table blocks between the two simultaneous chromosomes. The fitness value of these Child chromosomes is again calculated similarly and then selected ones are again sent for crossover, this process continues over a few generations. The child chromosomes of the last generation are expected to have good fitness values the results generated from which can be displayed.

If a chromosome has fitness value very near to the goal fitness value then that chromosome can be mutated to reach the goal value. The probability of any chromosome undergoing mutation is very less and almost negligible. In the system if any chromosome undergoes mutation then only the alignment of any of the object is changed to reach the goal value. To decide which furniture object's alignment will change the difference between its goal fitness value and current fitness value is calculated. The alignment of object with maximum difference is then changed.

Object undergoing change in alignment = Object with max (difference (goal fitness value, current fitness value)

= Max ((70-Wfit), (60-Bfit), (50-Tfit))

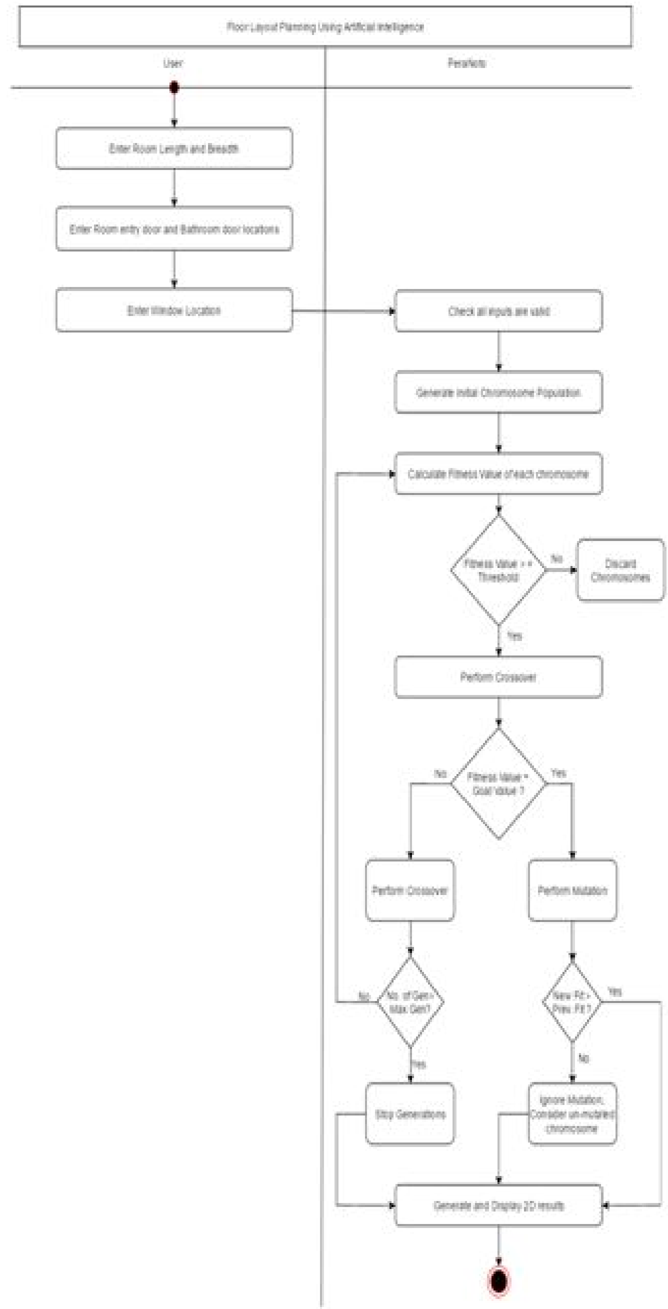
70 - Wfit = 4			60 - Bfit = 5			50 - Tfit = 7		
X_B	Y_B	1	X_T	Y_T	2	X_W	Y_W	1
1	2	3	4	5	6	7	8	9

In the above image since the difference between 50 - Tfit is maximum thus the alignment of Table is changed during Mutation. Thus after mutation the chromosome changes to the below image

X_B	Y_B	1	X_T	Y_T	2	X_W	Y_W	2
1	2	3	4	5	6	7	8	9

The mutation does not guarantee that the resulting chromosome will have a fitness value greater than its previous value. Hence, the fitness value is again calculated. If the new fitness value is greater than the previous value only then the mutated chromosome is included in the result otherwise the previous un-mutated chromosome is part of the result.

V. Activity Diagram



VI. Result and Analysis

The different evaluation parameters can be analyzed in the following manner.

· **Relevance:**

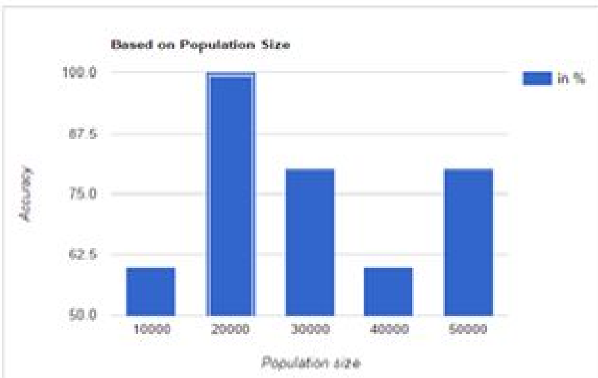
The analysis of relevance of a system can be explained as: the proposed system is relevant as it is one of its kinds. Being an NP-Complete problem it hasn't been touched by people yet and so this system gives scope for future development on this system as well as other similar systems by using this approach.

- **Efficiency, effectiveness and accuracy:**

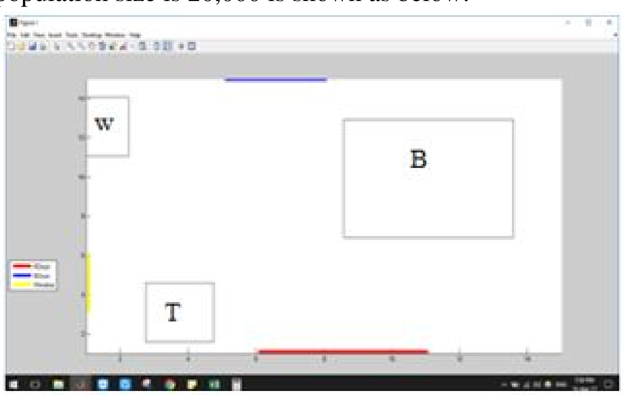
Analysis of the efficiency, effectiveness and accuracy of Genetic algorithm depends mostly on two factors:

1. Initial population size
2. Number of generations

Depending on the particular requirement, an optimum size has to be selected. When the values of accuracy for a population size were plotted on a graph, it was observed that as the size of the initial population was 20,000 the system generated optimum results as compared to other sizes.



In each of the output windows below, Rectangle B represents Bed, Rectangle W represents Wardrobe, Rectangle T represents Table. One of the outputs when the population size is 20,000 is shown as below:

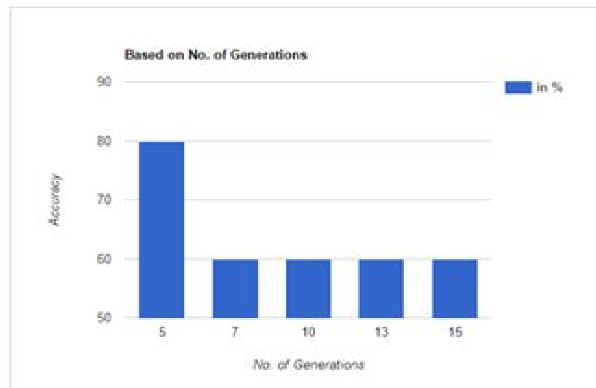


The above result follows all the mentioned constrained and also produces an aesthetically pleasing arrangement since it places the Wardrobe and Table nearer to the walls and the Bed is also positioned in a relatively good position

Comparably, the output for other sizes is not as satisfactory:

Though the above image follows all the constraints of furniture arrangement applied to the system, the output generated is almost repetitive with hardly any appreciable difference in the results. Also, the Wardrobe is placed very near to if not in front of the window which is not very desirable.

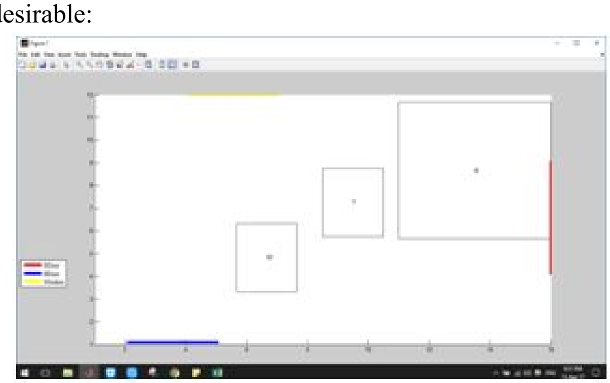
On varying the number of generations, we observed that best outputs were received for 5 generations. In genetic algorithm, if the system runs out of chromosomes mid-generation then the rest of the generations are useless computations.



One of the outputs received when number of generations was kept 5 is as follows:

The above image gives an acceptable result of the Bed being in almost center of the room while the Wardrobe is in the corner or adjacent to the wall and Table is placed nearer to the Bed.

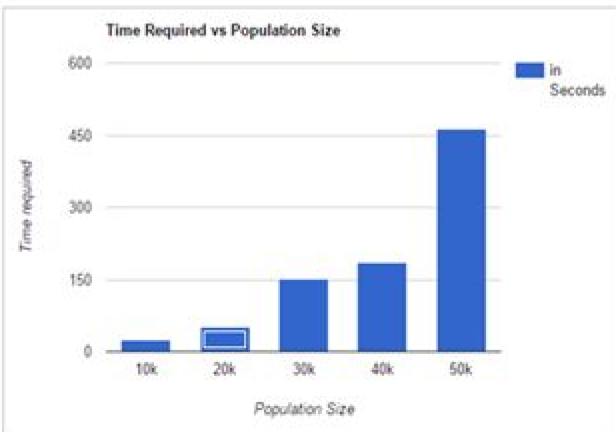
Comparably, for other cases, the output is not as desirable:



The above result places Wardrobe in the center of the room which is not desirable

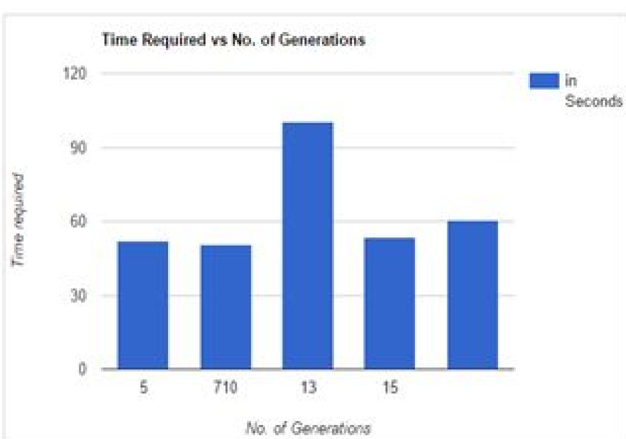
Based upon above observations, for our system, the number of generations has been decided as 5, for an initial population size of 20,000.

The relation of population size with time required is directly proportional. Upon varying the population size, the time required for the system to generate an output increases exponentially. The graph of time required vs. population size is as follows:



As the size of the initial population increases, the system has to deal with more genes, resulting in increased computations required for processing the genes.

However, there doesn't seem to be any discernible relation between the time required and number of generations. There were no significant changes observed as the number of generations was increased or decreased, affirming that the time required for this system to generate a solution solely depends on the population size.



VII. Conclusion

PeraNoto uses genetic algorithm, a technique under the shade of artificial intelligence, to produce furniture layout plans for the user. PeraNoto requires basic input that includes size of the room followed by the position of doors and window from the any wall as per the user's convenience. These types of layout generations can be extended to include the scope of township planning, motion sensor placement in smart environments and in the generation of realistic indoor scenes in floor planning. The system developed is efficient and an extension of it would also be possible and not a problem.

VIII. Future Scope

A. Township Planning

The concept of planned optimized arrangement of objects can be applied in the case of Township planning. Using Genetic Algorithm (GA) the arrangement of residential buildings keeping in mind their proximity to schools, markets and hospitals can be planned. It can also be used to plan the layouts of roads and bridges as well as various connecting lanes in a township. Layout of various public places can also be planned in similar way.

B. 3D Output

In the proposed system the generated output will be represented in 2D or as text format. A more convenient representation of the output for the user would be a 3D visualization of the furniture elements in their different arrangements. This can be achieved by using augmented reality concepts for generating 3D outputs.

C. Interior Designing

The concept of floor layout planning with the help of genetic algorithm can be extended to a broader scale when planning interiors of a whole office or a luxury hotel. Same principles that will be applied in the proposed system can be modified according to the size of the place to be designed and number of elements to be considered.

Acknowledgment

We remain immensely obliged to Prof. Varunakshi Bhojane for her continuous help and invaluable support in garnering resources for us. It is her guidance and supervision which made this project happen. We also express our gratitude to Prof. Ranjita Gaonkar for the inspirations at the beginning of our project and also for giving her enlightening comments till the end of it. We would also like to thank the H.O.D. of Computer Department, Dr. Madhumita Chatterjee for her invaluable support throughout the project. We would also like to thank our principal Dr. R.I.K.Moorthy for giving us this opportunity and trusting us to build a project that has no existing system.

References

[1] Jun H. Jo and John S. Gero, "Space Layout Planning using an Evolutionary Approach", Artificial

Intelligence in Engineering, Volume 12, Issue 3, July 1998, pg. 149-162

[2] Kristian Guillaumier, "Generic Chromosome Representation and Evaluation for Genetic Algorithms", Proceedings of Computer Science Annual Workshop (CSAW '03), November 2003, pg. 64-67

[3] MohdSaberi Mohammad, Safaai Deris, Safie Mat Yatim, Muhammad Razib Othman, "Feature Selection Method Using Genetic Algorithm for the Classification of Small and High Dimension Data", First International Symposium on Information and Communications Technologies, October 2004

[4] Ryota Tachikawa and Yuko Osana, "Office Layout Support System using Genetic Algorithm– Generation of Room Arrangement Plans for Polygonal Space", International Symposium on Nonlinear Theory and its Applications, September 2010

[5] Eugénio Rodrigues, Adélio Rodrigues Gaspar and Álvaro Gomes, "An evolutionary strategy enhanced with a local search technique for the space allocation problem in architecture, Part 1: Methodology", Computer-Aided Design, Volume 45, Issue 5, May 2013, pg. 898-910

[6] Brian L.Thomas, Aaron S. Crandall and Diane J. Cook, "A Genetic Algorithm approach to motion sensor placement in smart environments", Journal of Reliable Intelligent Environments, Volume 2, Issue 1, April 2016, pg. 3-16

[7] Lap-Fai Yu, Sai-Kit Yeung and Chi-Keung, "Make it Home: Automatic Optimization of Furniture Arrangement", ACM Transactions on Graphics (TOG) – Proceedings of ACM SIGGRAPH, Volume 30, July 2011, article no.86