RESEARCH ARTICLE

FLOOR LAYOUT OPTIMIZATION USING GENETIC ALGORITHM

*Nitin Choubey

Department of Computer Science & Information Technology, SVKM's NMIMS University, MPSTME, Shirpur, India

ARTICLE INFO	ABSTRACT
	This papers present a floor layout optimization system with Genetic algorithms. Efficient market floor layour management is required for better customes satisfaction and increased visibility of the product. It can be achieved through propoer allocation of the shelves on shop floor. The paper also compares the results for layout optimization for different sizes with the on varying population size.

INTRODUCTION

Facility planning is arrangement of the goods, services, shelves for production of goods or for delivery of the services (Figure 1). The main objective of the facility planning is to bring our maximum customer satisfaction. Research (Juel-Jacobsen, 2015) on pathway design has shown that well-established principles of urban retail designs are very important for retail managers, in particular for supermarkets and larger retail stores. The store layout influences both shopping atmosphere and shopping behavior of consumers visiting the store (Lewison, 1994). A well designed store layout can contribute to a positive shopping atmosphere, which results in the kind of shopping behavior, a retailer wants to achieve. According to study by Timjen Elbers (2016), Grid layout, long pathways which are placed parallel to each other, facilitates an efficient and fast shopping experience. Retailers are in favor of grid layout style because the rectangular arrangement of the shelves fits well in the shopping behavior of consumers and has universally the most preferred layout style by supermarket retailers. Tompkins and White (1996) discuss about 50% of total operating expenses can be reduced through a good placement of facilities. John Holland invented GA in 1960s (Holland, 1975). In contrast with evolution strategies and evolutionary programming, Holland's original goal was not to design algorithms to solve specific problems, but to formally study the phenomenon of adaptation as it occurs in nature and

Corresponding author: Nitin Choubey,
Department of Computer Science & Information Technology, SVKM's NMIMS University, MPSTME, Shirpur, India.

to develop ways in which the mechanisms of natural adaptation might be utilized into computer systems. Holland (1975) in 'Adaptation in Natural and Artificial Systems' presented the GA as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA. Many problems in engineering and related areas require the simultaneous genetic optimization for a number of possibly competing objectives. These have been solved by combining the multiple objectives in to single scalar by the approach of linear combination. The combining coefficients, however, usually based on heuristic or guesswork can exert an unknown influence on the outcome of the optimization. A more satisfactory approach is to use the notion of Pareto optimality by Goldberg (1989) in which an optimal set of solutions prescribe some surface 'The Pareto front' in the vector space of the objectives (Goldberg, 1989). For a solution on the Pareto front no objective can be improved without simultaneously degrading at least one other. This is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), generates solution to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0's and 1's, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every

individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. Genetic algorithm (GA) has been widely used in optimization with binary and continuous variable and is popular in solving facility layout problem, R. Haupt and Haupt (2004). GA are found to be efficient search methods for finding solution of Travelling Salesperson problem (Choubey, 2012; Choubey, 2013), Travelling Tournament problem (Choubey, 2010), Solving different maze structures (Choubey, 2012), grammar induction (Choubey and Kharat, 2012; Nitin Choubey and Madan Kharat, 2011; Choubey and Kharat, 2010) etc. Misola and Navarro (2014) presented genetic algorithm to solve facility layout problem to minimize total cost function (Maricar G. Misola and Bryan B. Navarro, 2014). Dalgic used GA for Market floor planning. The objective of the paper is to find the optimal grid-shelf layout for the floor planning using Genetic Algorithm with different level of complexity for the specific rectangular floor dimension.

Figure 1. Typical Market shop floor (Maricar G. Misola and Bryan B. Navarro, 2014)

Methodology Used

The following subsections introduce the various important components used in the genetic algorithm process.

Genetic Algorithm

The GA produces successive generations of individuals, computing their "fitness" at each step and selecting the best of them, when the termination condition arises. Genetic Algorithm approach used in the papers is given below.

1. Create initial random population.
2. Calculate fitness of the individuals in the population.

3. Perform selection of the individual by Tournament selection method.
4. Generate offspring by single point crossover.
5. Generate offspring by swap bit mutation.
6. Calculate fitness for the generated offspring.
7. Combine offspring generated in step 4 & step 5 with current population to get next population by overlapped population method.
8. Repeat steps 3 through 7 until termination criterion is reached.

The offspring are generated separately by the process of crossover and mutation and the entire collection is treated as child population. The resultant child populations is them combined with parent population to get next population.

Chromosome structure and Mapping

The chromosome structure encoding uses decimal numbers for representing the cells for shelves placement. Figure gives the chromosome representation and the equivalent for the chromosome representation. To get the optimal length of the chromosome, it is kept at size of twice that of the number of racks (Shelves). The chromosome bit is encoded as 4-tuple {Row, Column, Direction, Length}. The codes for direction are 0, 1, 2 & 3 for arrangement of the shelves from row-column position to Right, Left, down and up respectively. The sample chromosomes structure for 10 shelves placed in rectangular layout of size 6 by 6 is shown in the Figure 2. The fitness of a chromosome is calculated based on several criteria's as given below.

1. No shelve should be placed in overlapping with another cell.
2. The ENTRY and EXIT cells should not be blocked.
3. No shelve should have more than three shelves kept adjacent to it.
4. The Grid layout should be followed.
5. There should not be isolated/unreachable areas from either ENTRY or EXIT in the layout.

The penalty is applied in case of each violation of the criteria mentioned above.

Experimental Setup and Result

The experiment is conducted JDK 1.6 on Experiment is done with JDK 1.4 on an Intel Core™2 CPU with 2.66 GHZ and 2 GB RAM. The experiment was examined for varying population size and the shelves to be placed in a 10 by 10 shop floor space. Maximum number of generation = 100000, Crossover Rate = 0.8 and mutation rate = 0.1 is used for the purpose of experiment. The length of chromosome is considered equal to the length double to the size of the number of racks to be placed. The Generation run is terminated, if the solution with desired fitness values is found before reaching to the maximum number of generations. Comparison of generation runs for the various numbers of shelves arrangements (25, 30, 35, 40) with different population size (10, 20, 30) are shown in the   Figure 3. As GA's works on initial random seed population, the collected data is represented as average of the best fitness over generation runs for ten executions samples for the purpose of comparison. The result obtained for the arrangement of 25, 30, 35 and 40 racks are shown in Figure 4.
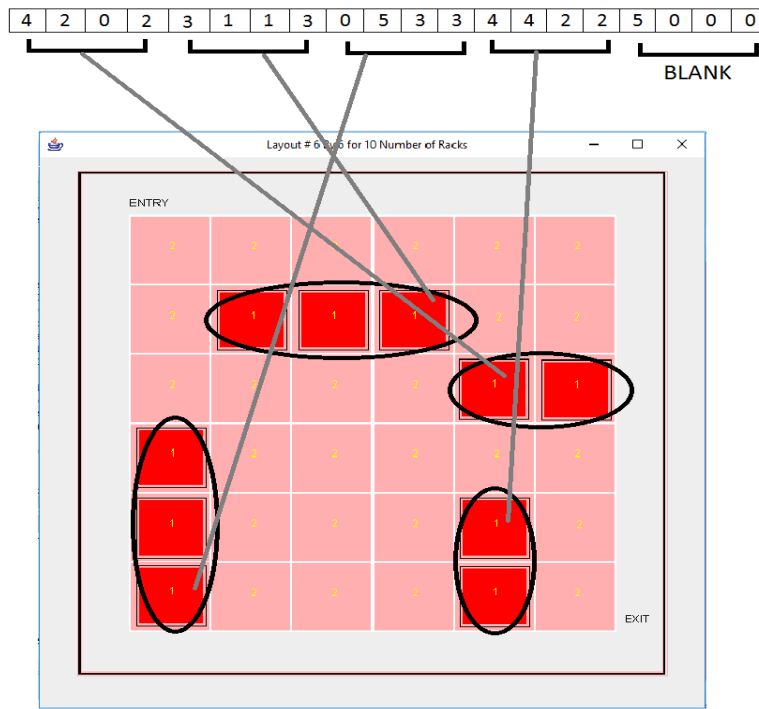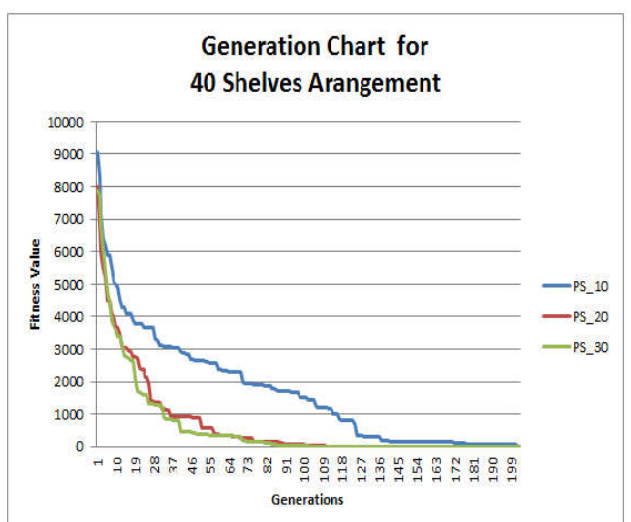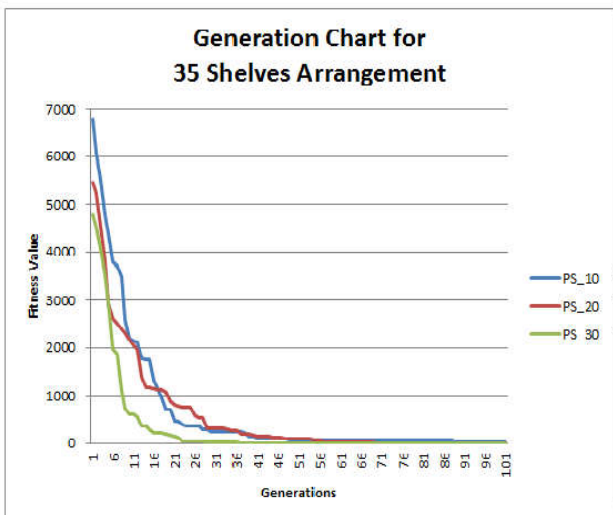
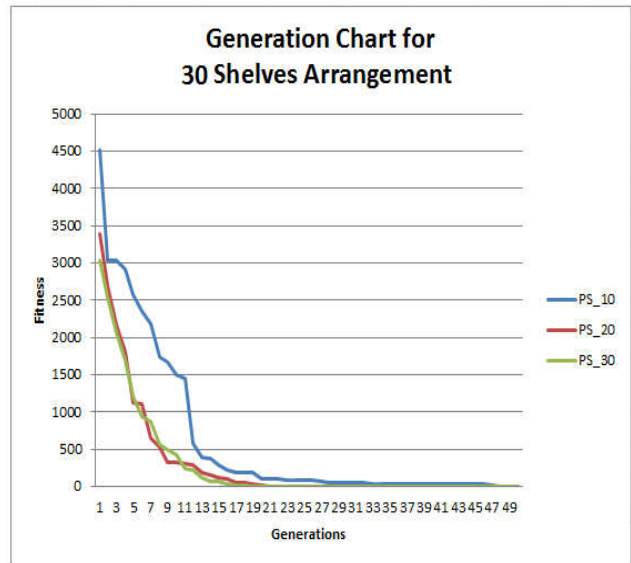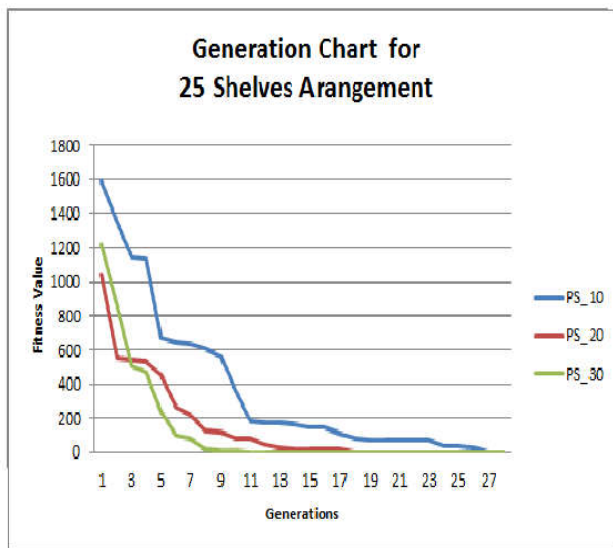**Figure 2. Chromosome structure and equivalent Layout**
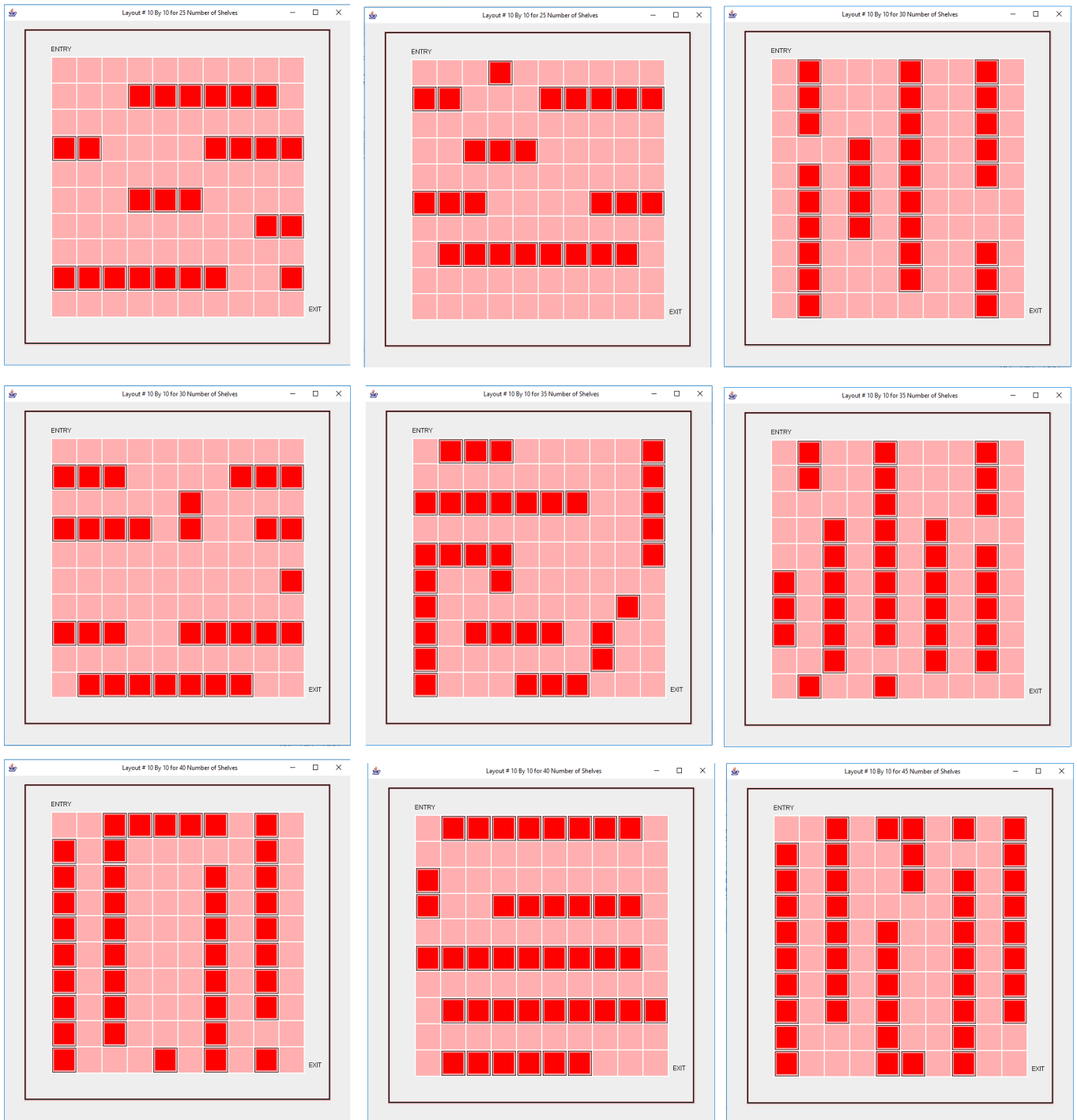

Figure 3. Generation Charts

Figure 4. 10 by 10 Layout results for 25, 30, 35, 40, 45 numbers of shelves

Conclusion

Genetic algorithms proved to be efficient method for finding the optimal results in market shop floor layout planning. It is found that Gas required higher number of generations to find optimal layout when used for higher number of shelves to be placed in the given floor dimension. Genetic Algorithms found to provide multiple optimal solutions to the given number of shelves requirement in the given area.

REFERENCES

Carl Heintz, How do you approach retail stores to sell your products?, August 2016 https://ceanow.org/marketing-sales/approach-retail-stores-sell-products/

Choubey N. S. 2010. A Novel Encoding Scheme for Traveling Tournament Problem using Genetic Algorithm, JCA Special Issue on Evolutionary Computation for Optimization Techniques (ECOT), *International Journal of Computer Application*, pp 79-82.

Choubey N. S. 2012. "A-Mazer with Genetic Algorithm", *International Journal of Computer Applications,* (0975-8887)58(17), November.

Choubey N. S. 2012. Solving TSP using DARO, *International Journal of Computer Applications*, (0975 – 8887) Volume 58– No.9, November 2012.

Choubey N. S. 2013. Moving Target Travelling Salesman Problem using Genetic Algorithm, *International Journal of Computer Applications*, (0975–8887) Volume 70–No.2, May.

Choubey N. S. and Kharat M. U. 2012. Hybrid system for handling premature convergence in GA – Case of grammar induction, Appl. Soft Computing J., http://dx.doi.org/10.1016/j.asoc.2012.03.069.

Choubey N.S. and Kharat M.U. 2010. "Grammar Induction Strategy Using Genetic Algorithm: Case Study of Fifteen Toy Languages". *Pacific Journal of Science and Technology*, 11(1):294-300.

Goldberg D.E. 1989. Genetic Algorithms in Search, Optimizations and Machine Learning, Addison-Wesley.

Haupt R. and S. Haupt 2004. *Practical Genetic Algorithms*, Second Edition, John Wiley and Sons, Inc.

Holland, J.H. 1975. Adoption in Natural and Artificial System, University of Michigan Press, Ann Arbor, MI.

Juel-Jacobsen, 2015. L.G. Aisles of life: outline of a customer-centric approach to retail space management, International Review of Retail, *Distribution and Consumer Research*, 25 (2), pp. 162-180

Lewison, D. M. 1994. Retailing. New York: Macmillan College Publishing Company.

Maricar G. Misola and Bryan B. Navarro, 2014. Optimal Facility Layout Problem Solution Using Genetic Algorithm, World Academy of Science, Engineering and Technology, *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, Vol:7, No:8.

Nitin Choubey and Madan Kharat, 2011. Approaches for handling premature convergence in CFG Induction Using GA, Gaspar-Cunha *et al*. (Eds.): Soft Computing in Industrial Applications, AISC 96, pp. 55–66, Springer-Verlag Berlin Heidelberg.

Timjen Elbers, 2016. The effects of in-store layout- and shelf designs on consumer behavior, http://edepot.wur.nl/369091

Tompkins J. A. and J. A. White 1996. *Facilities Planning*. 2nd Ed., New York, John Wiley.

****