IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 5, July 2010
ISSN (Online): 1694-0784
ISSN (Print): 1694-0814

Shape Assignment by Genetic Algorithm towards Designing Optimal Areas

Ismadi Md Badarudin[1], Abu Bakar Md Sultan[2], Md Nasir Sulaiman[2], Ali Mamat[2], Mahmud Tengku Muda Mohamed[3]

[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA
Shah Alam, Selangor, Malaysia

[2]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia
Serdang, Selangor, Malaysia

[3]Faculty of Crop Science, Universiti Putra Malaysia
Serdang, Selangor, Malaysia

Abstract

This paper presents a preliminary study on space allocation focusing on the rectangular shapes to be assigned into an area with an intention to find optimal combination of shapes. The proposed solution is vital for promoting an optimal planting area and eventually finds the optimal number of trees as the ultimate goal. Thus, the evolutionary algorithm by GA technique was performed to find the objective. GAs by implementing some metaheuristic approaches is one of the most common techniques for handling ambiguous and / or vast possible solutions. The shape assignment strategy by the determined shapes coordinate to be assigned into an area was introduced. The aim of this study is to gauge the capability of GA to solve this problem. Therefore some strategies to determine the chromosome representation and genetic operators are essential for less computational time and result quality. Some areas coordinate were used to generate the optimal solutions. The result indicates the GA is able to fulfill both feasible result and acceptable time.

Keywords: Genetic algorithm, Shape assignment, Space allocation, Optima solution, Metaheuristic

1. Introduction

1.1 Designing Optimal Layout

Refer to lining layout planning (LLP) [1], the optimal design with the strategy of determining area division by blocks and then choosing the best line direction for each block was introduced. The different blocks basically produces the different number of trees, thus the right chosen of blocks is vital for area optimization will eventually promoting optimal number of trees to be planted in an area. We have managed by exact method in order to find the best line direction in an area due to lines that differentiated by number of degree (between 1^0 and 90^0) from baseline has 90 possibilities to be analyzed. Meanwhile deciding the blocks, we employed a Shape Assignment strategy. However, it could be considered as an ambiguous matter with many possible solutions requires an exponential time to reach a solution. To handle this, some metaheuristic approaches by Genetic Algorithm (GA) implementation were applied.

Optimal layout means the combination of shapes promote as less as unused space. In LLP, we attempt to find optimal solution whenever the combination of shapes must completely be fitted with no unused spaces as shown in figure 1. The existence of unused space or / and at least one shape beyond the border of area will be rejected because of by respective with area is not fully utilized and violation of hard constraint. The solution will be used for assigning the appropriate line direction for each determined shape will consequently achieve the optimal tree density.

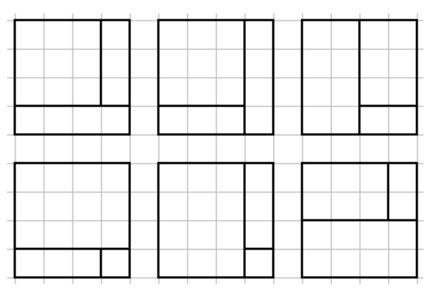

Fig.1 Possibility of shapes that represent blocks to be assigned into an area

In order to divide an area to blocks, the determination of shapes pattern with their possible coordinates. The number of shapes to be suited into area depends on the number of blocks that is determined by user. The blocks that represented by shape patterns (square, vertical or horizontal rectangular) with various coordinates were used to be assigned into rectangular areas as shown in figure 2.

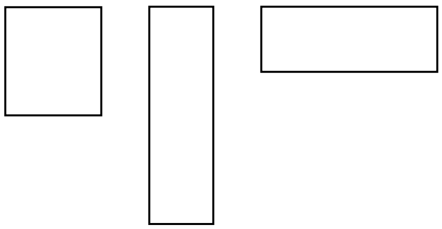

Fig.2 Shape patterns

The coordinates of area and shapes refer to top-right (x4, y4), while bottom-left (x1, y1) is permanently set with (0, 0). The possible shapes coordinates to be assigned into area up to the area coordinate. For example, the rectangular area of 4, 5 coordinate means (x1, y1), (x2, y2), (x3, y3) and (x4, y4) represented by (0, 0), (4, 0), (0, 5) and (4, 5) respectively as shown in figure 3. The possibility of shapes to be combined if the area coordinate of (4, 5) are (1, 1), (1, 2), …, (4, 4), (4, 5) as shown in figure 4. The size of area is acquired by multiplication of x4 and y4.

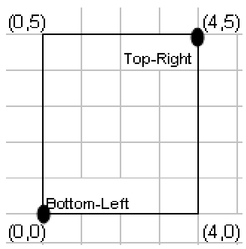

Fig.3 Representation of (4, 5) area coordinate

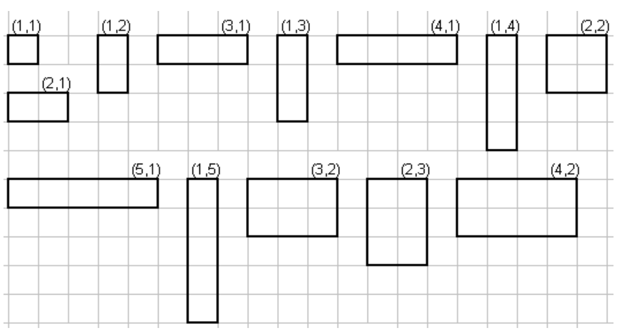

Fig. 4 Possible shapes coordinates refer to the area coordinate of (4, 5)

This paper only devotes the Shape Assignment strategies of GA implementation towards searching the optimal shapes combination in an area. This problem relates to Space Allocation Problems (SAPs) that has uncertainty solutions; we found the metaheuristic is the appropriate approaches to be applied as discussed in section 1.2. In order to overcome this issue we organize this paper in the following ways. Section 2, the brief explanation of GA as the selected algorithm, followed by the GA strategies used in Shape Assignment solution in section 3. An empirical analysis on GA capabilities for handling this issue and the results were reported in section 4. Finally we conclude in section 5.

1.2 Space Allocation by Metaheuristic Approaches

Rectangular Shape Assignment into an area is one of the challenges in Space Allocation Problems (SAPs). The ambiguity solution to determine the appropriate shapes to be fitted into an area is considered as a type of optimization problems. The aim is to minimize unused space so that the area could be optimized. The searching processes to meet the objective function, however, promote many possible solutions to be analyzed in accordance with the available constraints and consequently lead to high computational time. SAPs are considered as Non-Deterministic Polynomial (NP) problem that requires exponential time to solve such related problems in the worst-case. In many cases, a problem can be solved by several algorithms and problem complexity is measured by the time complexity of the most efficient algorithm for the problem [2]. A tractable problem has certain solution to be found and it might employ exact method techniques that can be solved in polynomial time. Although these techniques could obtain optimal solution, it can be computationally expensive and impractical for many real world applications. Whereas the problems are considered as intractable require probabilistic techniques with guide intelligently in search space. The intractable problems are unsolvable by exact methods since the computational time is exponential. According to Ruibin [3] Polynomial (P) represents by class of the problems that are solvable by deterministic algorithm with polynomial time complexity. Non deterministic Polynomial (NP) is the class problem that can be solved in polynomial time by a non-deterministic algorithm.

Some solutions in SAP by exact methods may found the required solution but the time computation is usually questionable. The approximate algorithms with metaheuristic approaches are usually proposed to solve this matter. The metaheuristic was introduced by Glover [1986] has been recognized to solve the problems with satisfactory solution quality within reasonable computational time in optimization problems. In spite of it does not guarantee to obtain the best solution but (near) optimal solutions are accepted. However, to find the optimal solution sometime fails in the middle of searching process because of getting trapped at the local optima is a main challenge. Some papers to avoid the premature convergence issue such as a maintaining diversity for GA strategy [4] and an improvement of the standard GA [18] have been proposed. The approaches of Metaheuristic have also been discussed by researchers [5, 16, 7], and then were simplified with nine properties [8]. Some properties such as 1) guide the search process by short or long term memory; 2) A strategy to find optimal solution; 3) Consolidation of basic search process and complex learning procedure make metaheuristic applicable and efficient to solve the optimization problems. Among algorithms that apply the metaheuristic approaches, GA is one of the most prominent and widely accepted for various types of optimization in many fields according to the number of literatures published. In order to find the efficiency and effectively of techniques, several comparative analysis by GAs, Simulated Annealing (SA) and Tabu Search (TB) have been organized. The remark that these techniques have their own strength and strategy to solve the optimization matters [9]. A satisfactory solution to a particular layout problem can be provided by the metaheuristic approaches such as Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS) that have promised to obtain better solution by proposing solutions [8]. Some papers related SAPs that focused on various fields such as architecture [10], retail shelf [3], fashion industry [11], computer graphic [12] employed GAs to handle their problems.

2. Brief Discussion of Genetic Algorithms (GAs)

Genetic Algorithms (GAs) are a technique that employs some properties of metaheristic. GAs are also a part of evolutionary computing by taking inspiration of evolutional theory that was expanded from Darwin's evolution thought to algorithm solution approach by Holland (1975). GAs are modeled on the natural process of gene evolution and restructure by repetition in which can find the global optimal solution in complex multi dimensional space search.

GAs are different from other heuristic methods in several ways. GAs are population-based method works on a population of possible solutions by mean the proposal of many solutions can be performed at one time, while other heuristic methods use a single solution in their iterations. Another difference is that GAs implement probabilistic (stochastic) and not deterministic approach to find solution. Even though GAs might have constraints by typically works at short-term memory and iteration for generating next generations makes time computation to find the solution is unpredictable, but a promising optimal solution with the strategy to protect from premature convergence make GAs one of the most selected techniques.

GAs have capability of handling the SAPs by 1) evolutionary strategy by genetic operators to propose new individuals in next generations that are most probable better solution, 2) mutation operator acts to avoid trap in local optimal stage. By random searching, the successful of finding solution is uncertainty where the use of long term memory (to store the history of movement and current solution) might be useful. Other strategies might improve the performance of GAs in terms of time and solution quality need to be applied. Determining population size, chromosome, problem representation, rate of crossover and mutation play the role to improve the searching process. GAs implementation begins with population initialization and followed by the three genetic operators.

The number of population must be cautiously clarified since population size plays the role of efficiency and effectiveness of searching process. Large population size has high possible to reach optimal solution but might increases the computational time per iteration. On the contrary, the number of generation might increase but the possibility of repetition of same solutions occurs is high as a result it might trap at local optimal. Several researchers have deployed the effect of these parameters of GA [3, 14, 15]. They conclude that a large population size means the simultaneous handling of many solution and increases the computation time per iteration; however the probability of convergence to a global optimal solution is higher when using a small population size.

3. GA in Shape Assignment Solution

3.1 Shape Assignment Solution

The objective of GAs is to find optimal solution of shapes combination that are put into rectangular area. The shapes are subject to their coordinates must comply with the three constraints as follow:

1. Shapes assigned must be fully fitted into the area. Therefore, the size of selected shapes must be equal to the size of area.
2. Assignment of shapes should start with the largest. This will avoid the same shapes combination at a time.
3. No same shape type is used at the certain time of assignment. Shape type refers to the use of shapes coordinate that produce shape size and pattern.

The strategies of shape assignment:

1. Placing of shapes based on the determined block number. For instance, if block number is four, shapes will be assigned will be four.
2. The combination of shapes type must be different for every solution.
3. The use of shapes to be assigned into area will be analyzed by considering item number of the shape. However the arrangement of shapes does not take into

account since it will produce the same number of item regardless where the shapes are placed.

The processes of Shape Assignment by GA begin with assigning value to genes of a chromosome (genome) that represents individual. The chromosome of population consists of shapes will be assigned into an area as figure 5. The chromosome that has fulfilled the constraints is considered as a successful individual; in contrast fail to find the successful individual new generation will be produced.

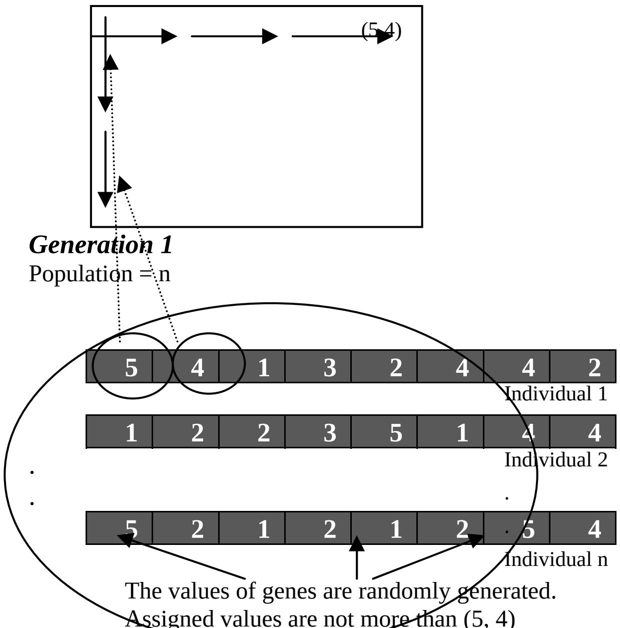

Fig.5 Shape Assignment Strategy

Figure 6 represents the successful solution with chromosome of 3, 3, 1, 4, 3, 1, all shapes were fully located into the area with 0 unused space. However, the figure 7 with 3, 3, 3, 2, 2, 2, can only be fitted one shape and it remains an unsuccessful solution.

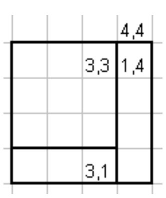

Fig. 6 Successful Solution

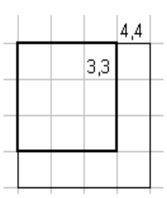

Fig. 7 Unsuccessful Solution

3.2 Problem Representation

The representation of problem by coding scheme requires specific consideration and observation because it has a major impact on the performance of GA. Different representation schemes might cause different performance in terms of accuracy and computation time [9]. The problems can be represented by bit or string. The typical encoding by bit representation, however, it is suitable for parameter take on an exact value for examples 128 = 7 bits, 1024 = 10 bits, otherwise unnecessary bit patterns may result in no evaluation [16]. The string representation may optimize the coding usage but it is less meaningful to represent the problems.

The Shape Assignment is represented by integer with regards to shape coordinates. The length of chromosome depends on two times of the block number (2 * B). The odd spaces of chromosome are represented by x4 coordinate, whereas the even spaces are y4 coordinate. To extend each adjacent odd and even is considered as x4 and y4 coordinate of a shape. We illustrate the design of chromosome as figure 8 below:

Block1		Block2		. . .		Block N	
Shape x4	Shape y4	Shape x4	Shape y4	…	…	Shape x4	Shape y4
2	1	4	5	…	…	3	4

Fig. 8 Chromosome represents X and Y coordinate of shapes

The area coordinate must be priory determined. To assign possible shapes into the area randomly, we developed a strategy of assigning the value into the genes depends on the x4 and y4 coordinate of determined area. For example, the x4 and x4 of area represented by 4 and 5 respectively, hence, the odd spaces of chromosome can be assigned a value between 1 and 4, similar to even spaces where a number between 1 and 5 is allowed. The possible solutions can be referred to chromosome value in figure 5. By this strategy, possible solutions will decrease by comparative illustration of figure 9 and 10.

To find the possible solutions the both simple random and our strategy are compared. Simple random assigns between 0 and 9 for both x4 and y4 employs formula ($10^{chromosomeLength/2} * 10^{chromosomeLength/2}$) = $10^4 * 10^4$ = 100,000,000 while by the strategy we manage (x4 areaCoordinate$^{chromosomeLength/2}$ * y4 areaCoordinate$^{chromosomeLength/2}$) = $4^4 * 5^4$ = 160,000 possible solutions. We found that, this strategy produces less possible solutions and significantly affect towards reducing the time processing.

2	5	1	2	1	2	4	5
0-9	*0-9*	*0-9*	*0-9*	*0-9*	*0-9*	*0-9*	*0-9*

Fig.9. Random number between 0 – 9

2	5	1	2	1	2	4	5
1-4	*1-5*	*1-4*	*1-5*	*1-4*	*1-5*	*1-4*	*1-5*

Fig.10. Random Strategy is based on X and Y coordinates of area

3.3 Fitness and Penalty Function

The constraints must be identified since the result of analysis relies on the satisfied constraints. Constraints can be classified as hard or soft. Hard constraints are those that cannot be violated while soft constraints are those that can be broken but penalized. To minimize the penalties in a solution for a space allocation problem, no hard constraints should be violated and as many as possible soft constraints should be satisfied [17]. In Shape Assignment strategy, the three hard constraints must be fulfilled to meet the optimal result.

Hard constraints:

1. The total size of selected shape type must be equal to area size. Area size = Shapes size.

$$areaX * areaY = \sum_{i=1}^{N} ((shapeX)_i * (shapeY)_i)$$

2. The use of shapes must be once for every solution.
3. Combination of shapes to be fixed into area with 0 unused spaces. The arrangement of shapes either row (R) or column (C) must take into account.

If R arrangement

$$\sum_{i=1}^{N} (shapeX)_i = areaX * 2$$

If C arrangement

$$\sum_{i=1}^{N} (shapeY)_i = areaY * 2$$

Soft constraints: will be given a weighted value to be penalized.

1. The largest size of first shape is better.

In order to find the acceptable shapes, a fitness and penalty function were implemented. This is because of to allocate the shape with zero unused space is our main consent to be achieved. Therefore, the value that reached the maximum value of fittest will be accepted. We assigned the fitness values for the constraints as table 1 and table 2 respectively. The successful individual is based on the value that reaches the value of maximum fittest (100). In table 1, the fitness value for successful condition of 1 and 2 are given by 20. While the condition 3 stated that all shapes are successfully fitted into area with zero unused space is assigned by 60. The fitness value of each fitted shape is based on 60 is divided by number of used shape. If only two out of three shapes number are successful the fitness value would be 40 (60 / 3 * 2). Whereas in table 2, fail to meet condition of soft constraints will be penalized 10 of total fitness value.

Table 1: Fitness Value

Conditions	Fitness Value
1. Combination of Shapes Size = Area Size	20
2. The use of shape types = 1	20
3. All shapes can be fitted into area	60

Table 2: Penalty Value

Conditions	Penalty Value
The largest size <> first shape	-10

3.4 Crossover

Crossover implementation occurs whenever no solution found at initial population. The changes value by crossover process will generate a new generation. The chosen parents to generate offspring based on the highest fitness. By crossover the two parents are chosen, the new generation produces the two new offerings as shown in figure 11. The strategy by two point crossover means the adjacent two genes (odd and even) are randomly selected from individual, and then the part of value from those two points is exchanged.

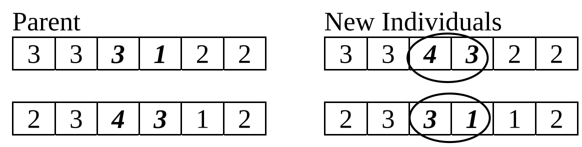

Fig.11 Process of crossover

3.4 Mutation

Mutation implementation occurs whenever no solution found at initial population and crossover stage. The changes value by mutation process is still in same generation. We assume that fitness reach at the accepted fitness level as mentioned in fitness function. Our assumption that the first and second genes represent the large shape, however the rest gene might not meet the acquired solution. With strategy randomly choose the two adjacent genes (odd and even) by ignoring the first and second gene. The random values, then to be assigned into the selected genes as shown in figure 12.

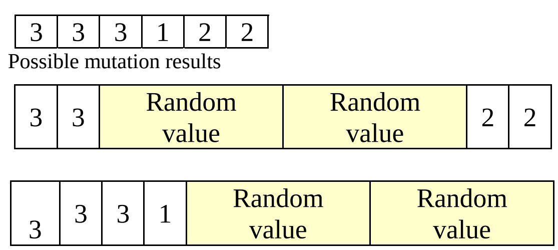

Fig.12 Process of mutation

4. Analysis and Result

We developed an application that employs GA strategies as mentioned in the previous section. The application provides flexibility for user to input area coordinate, number of block and some parameters of GA as shown in figure 13. According to our aim is to determine the time and result generated by this application, several series of experiment by different inputs were carried out as shown in table 3. A proposed application by GA was run using the computer performance of Intel Pentium processor 1.7 GHz, 400 MHz FSB and 2 MB L2 cache memory. Then the results were stated to be analyzed as shown in table 4.

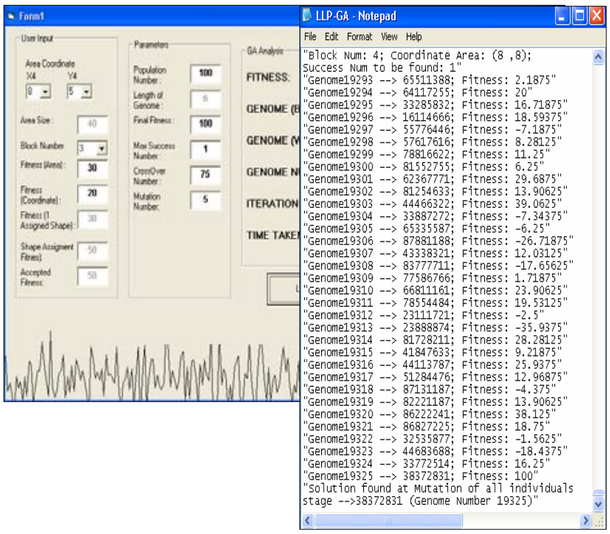

Fig.13 Example of application by GA strategies to generate optimal solution

4.1 Datasets

We conducted the series of experiment with different coordinates that represent the areas. For discussion purposes, we provide the four experiments that have been conducted. Each experiment has x4, y4 coordinate, scale and block number were input by user. The actual area coordinate and size were based on x4, y4 coordinate multiply by determined scale and actual area coordinate of x4 multiply by y4 respectively. The datasets are shown in table 3.

Table 3: Datasets for four experiments

Exp. Number	x4, y4 coordinate	Scale	Number of block	Actual Area coordinate (x4, y4)	Size (Hectare)
1	(4, 4)	100	3	(400, 400)	16
2	(8, 5)	100	3	(800, 500)	40
3	(8, 8)	100	4	(800, 800)	64
4	(3, 3)	100	4	(300, 300)	9

4.2 Result

We executed the application two times for every experiment. Some information generated by the application were stated such as the shapes combination that promotes zero unused space, number of iteration and taken time to be analyzed. The results are shown in table 4.

Table 4: Optimal Solution Analysis

Exp. Num	Run Num	Shapes Coordinate	Actual Blocks Coordinate	Time (sec)	Iteration Num
1	1	4, 2 3, 2 1, 2	400, 200 300, 200 100, 200	**2.40**	**2110**
	2	3, 3 3, 1 1, 4	300, 300 300, 100 100, 400	**1.74**	**1200**
2	1	4, 5 4, 1 4, 4	400, 500 400, 100 400, 400	**3.80**	**3500**
	2	5, 5 3, 2 3, 3	500, 500 300, 200 300, 300	**5.94**	**5300**
3	1	8, 6 5, 2 1, 2 2, 2	800, 600 500, 200 100, 200 200, 200	**13.5**	**15710**
	2	8, 4 7, 2 1, 2 8, 2	800, 400 700, 200 100, 200 800, 200	**29.9**	**21947**
4	1	Not Successful			
	2	Not Successful			

From the above result, we found the three important matters to be highlighted:

1. The taken time was not consistent and it was difficult to be expected even the use of same number of block and x4, y4 coordinate. This is because of the probabilistic and randomness strategies in GA.
2. The experiment showed the increase of time computation when the number of block and x4, y4 coordinate increase. The reason is the chromosome length is based on number of block and the values of genes are based on x4, y4 coordinate, as a result the more possible solutions to be analyzed.
3. In experiment 1 to 3 produced successful combination of shape that fulfills the constraints. However, the experiment 4 fails to propose the result when the premature convergence took place. This situation occurs because of the unsuitable x4, y4 coordinate consequently the existence of two or more same shapes cannot be avoided.

We conclude that the GA by shape assignment strategy is able to promote the optimal solution by referring to time and quality of solution. Even thought the time will increase when the number of block and size of x4, y4 coordinate raise, however the time is relatively small. Meanwhile, the successful results will be certainly proposed if the appropriate input of x4 and y4 coordinate.

5. Conclusion

The Shape Assignment with GA technique has promised to obtain the optimal solution. However, this study refers to the areas of square and rectangular shape and it become more complicated for trapezoidal areas in which a variety of areas coordinates must be accurately determined. Our future direction is to conduct a study in order to provide the solution strategies.

This effort is the first attempt to facilitate planner for designing optimal layout. The successful strategy of utilized space by GA techniques is a preliminary stage towards obtaining the optimal tree density in plantation areas is our ultimate aim. To achieve this, we are conducting the optimization strategy by combining GA and exact method to obtain optimal layout solution.

Acknowledgments

This research is registered under the Fundamental Research Grant Scheme (FRGS) with grant number is FRGS/03-04-10-873FR/F1 which is fully funded by the Ministry of Higher Education (MOHE), Malaysia. The authors express our great appreciation and thanks to MOHE, Malaysia and Universiti Teknologi MARA (UiTM), Malaysia for sponsoring one of the authors for in PhD research. The authors also wish to thank the Universiti Putra Malaysia (UPM) which provided the facilities and favorable environments for carrying out this research.

References

[1] Ismadi Md Badarudin, Abu Bakar Md Sultan Md Nasir Sulaiman., Ali Mamat., Mahmud Tengku Muda Mohammed (2010), Preliminary study on Designing Optimal Layout by Metaheuristic Approaches for Optimizing Planting Area, *International Journal Electrical and Computer Science IJENS-IJECS ,10* (1), pp. 92-96.

[2] Garey, M.R. and Johnson, D.S., (1979), Computer and Intractability: A Guide to the Theory of NP-Completeness, *W.H.Freeman*

[3] Ruibin Bai (2005), An Investigation Of Novel Approaches For Optimising Retail Shelf Space Allocation, *PhD Dissertation*, pp. 4, 31- 35

[4] Abu Bakar Md. Sultan, Ramlan Mahmod, Md. Nasir Sulaiman & Mohd. Rizam Abu Bakar (2006), Maintaining Diversity for Genetic Algorithm: A Case of Timetabling Problem, *Jurnal Teknologi*, 44(D) Jun 2006, pp. 123–130

[5] Osman, I.H. And Laporte, G. (1996), Metaheuristics: A bibliography. *Ann. Operational. Research 63*, pp. 513–623.

[6] Stutzle, T. (1999), Local Search Algorithms for Combinatorial Problems - Analysis, Algorithms and New Applications. *DISKI - Dissertation* Zur K Unstliken Intelligenz. In fix, Sankt Augustin, Germany.

[7] Voß, S., Martello, S., Osman, I. H., And Roucairol, C., Eds. (1999), *Meta-Heuristics—Advances and Trends in Local Search Paradigms for Optimization*. *Clawer Academic Publishers*, Dordrecht, The Netherlands.

[8] Cristian Blum and Andre Roli (2003), Metaheuristic In Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Journal,* pp. 268 - 307

[9] D.T.Pham and D.Karaboga (2000), Intelligent Optimisation Techniques: Genetic Algorithm, Tabu Search, Simulated Annealing and Neural Network, Book*: Springer*, pp. 1-50

[10] Mecklenburgh, J. C. (1985), Process Plant Layout. London: *Institution of Chemical Engineers*

[11] J. Martens (2004), Two genetic algorithms to solve a layout problem in the fashion industry, *European Journal of Operational Research*, 154 304–322

[12] Ads by Google (2009), Method for optimizing space allocation in table layout, [Online] Available: http://www.patentstorm.us/patents/7500179/description.htm

[13] Fogarty, T.C, (1989), Varying the probability of mutation in the genetic algorithm, *Proceeding 3rd Int. Conf. on Genetic Algorithm and Their Application*, George Mason University, pp. 104-109

[14] Grefenstette. J. J, (1986), Optimisation of control parameter for Genetic Algorithm, *IEEE Trans. On System, Man and Cybernetics*, VolSMC-16, No 1, pp 122-128

[15] Schaffer, J.D., Caruana, RA Eshelman, L.J. and Das, R, (1989), A study of control parameters affecting on-line performance of genetic algorithm for function optimisation, Int*. Conf. on Genetic Algorithm and Their Application*, George Mason University, pp. 51-61

[16] Darell Whitley, A Genetic Algorithm Tutorial, *Paper for Computer Science Department, Colorado State University Fort Collin*.

[17] Jesus Dario Landa Silva (2003), Metaheuristic and Multiobjective Approaches for Space Allocation, *PhD Dissertation*, page 14, 45.

[18] J. Andre and P. Siarry (2001), T. Dognon, An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization, *Advances in Engineering Software 32 Journal,* pp. 49 – 60.

Ismadi is attached as lecturer to the Computer and Mathematical Sciences Faculty at UiTM, Malaysia. He started as a lecturer in 2001 and was appointed as a senior lecturer in 2008. He obtained a Master's Degree in 2002 in metaheuristic Approaches in Timetabling. He is currently on leave as a lecturer and pursuing a PhD in Computational Intelligence at UPM, Malaysia under the supervision of Abu Bakar.

Abu Bakar is a senior lecturer holding a doctorate in the faculty of Computer Science and Information Technology, UPM. His fields of expertise are Metaheuristic and Evolutionary Computing. HE has published several journal papers regarding metaheuristic approaches have been published since 2004.

Nasir and **Ali** are Associate Professors holding doctorates in the faculty of Computer Science and Information Technology, UPM. Nasir has expertise in Intelligent Computing, Intelligent Software Agents and Data Mining, and Ali is expert in Database and XML. **Mahmud** is an Associate Professor and PhD holder in Plant Physiology/ Post Harvest and an Associate Professor in the faculty of Crops Sciences. His areas of expertise are Agricultural Sciences, Plant Physiology and Post Harvest Physiology.