

AN IMPROVED GENETIC ALGORITHM WITH GENE VALUE REPRESENTATION AND SHORT TERM MEMORY FOR SHAPE ASSIGNMENT PROBLEM

Ismadi Md Badarudin

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

Abu Bakar Md Sultan, Md Nasir Sulaiman, Ali Mamat

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

Mahmud Tengku Muda Mohamed

Faculty of Crop Science, Universiti Putra Malaysia, Serdang, Selangor, Malaysia

SCITEPRESS

Keywords: Genetic algorithm, Specific-domain initialization, Short term memory, Shape assignment.

Abstract: The purpose in shape assignment is to find the optimal solution that combines a number of shapes with attention to full use of area. To achieve this, an analysis needs to be done several times because of the different solutions produce dissimilar number of items. Although to find the optimal solution is a certainty, the ambiguity matters and huge possible solutions require an intelligent approach to be applied. Genetic Algorithm (GA) was chosen to overcome this problem. We found that basic implementation of Genetic Algorithm produces uncertainty time and most probably contribute the longer processing time with several reasons. Thus, in order to reduce time in analysis process, we improved the Genetic Algorithm by focusing on 1) specific-domain initialization that gene values are based on the X and Y of area coordinate 2) the use of short term memory to avoid the revisit solutions occur. Through a series of experiment, the repetition of time towards obtaining the optimal result using basic GA (BGA) and improved GA (IGA) gradually increase when size of area of combined shapes raise. Using the same datasets, however, the BGA shows more repetition number than IGA indicates that IGA spent less computation time.

1 INTRODUCTION

Genetic algorithm (GA) is an alternative method of solving many design problems which are considered as ambiguous problems and / or consists of the huge possible solutions. The initial step is to represent a legal solution to the problem to be solved by a string of genes that can take on some value from a specified finite range or alphabet. This string of genes, which represents a solution, is known as a chromosome. Each chromosome represents a legal solution to the problem and is composed of a string of genes. The binary alphabet $\{0, 1\}$ is often used to represent these genes but sometimes, depending on the application, integers or real numbers are also used. In fact, almost any representation can be used

that enables a solution to be encoded as a finite length string.

The GA's chromosome consists of groups of variables, which are represented by groups of genes. The initialization of these genes significantly affects the GA's performance, and an improper choice for the chromosome structure will often result in poor performance. The good representation of a chromosome explains the power of the GA search because they improve its efficiency and effectiveness. The better a chromosome contribute, the more of its genes will be preserved for the next generation (Chen-Fang Tsai and Kuo-Ming Chao, 2007). The good coding for a chromosome representation will ensure reduce the possible number of proposed solutions to be analyzed. According to Schaffer (1985) the short defining of

chromosome length consisting of bits which work well together, and lead to improved performance when incorporated into a chromosome.

The reasons using GA because of the metaheuristic properties that are applied in GA to reach the optimal result in term of time and quality are better (Goldberg, 1989; Miihlenbein and Schlierkamp, 1993; Srinivas and Patnaik, 1994). With some properties of metaheuristic are implemented by GA promise better solution; however certain situations might need an improvement of GA with several strategies to yield an efficient time.

One of the nine metaheuristic properties is strategies to guide the search process. Many of the metaheuristic approaches rely on probabilistic decision made during the search. But, the main difference to pure random search is that in metaheuristic algorithms randomness is not used blindly but in an intelligent, biased form (Stutzle, 1999). Such strategy allows generating new starting solutions for the local search in more an intelligent way rather than just providing random initial solutions. In the evolutionary computational field (Eiben and Schippers, 1998), the process of exploration and exploitation often refer to short term strategy tied to randomness. According to Blum and Roli (2003), a short term memory to escape from local minima and to avoid cycles, this property is often applied in simple tabu search. In addition, the different approaches to other methods to solve trajectory optimisation by applying the random value with evolutionary strategy that has global search capability and the robust characteristics (Rae-Dong Kim et al., 2007). Nevertheless, their efficiency has recently been criticized because the repetition number in GA influences the length of computation time in analysis process.

This paper focuses on the strategies of specific-domain initialization rather in assigning value of genes and to avoid revisit solution occurs. We divide this paper to 5 sections. The reminder of this paper is organized as follows, in section 2 is the overview of application problem to be settled with the constraints. In section 3 devotes to discussion of the two improvement strategies that were applied in basic GA (BGA) named by improved GA (IGA). In chapter 4 reveals the results based on analysis of generating solutions and followed by discussion on the results. Finally in section 5, we discuss the conclusion of this paper.

2 OVERVIEW OF SHAPE ASSIGNMENT PROBLEM

Shape assignment purpose is to combine the several of shapes to be assigned into a determined area. The objective is to fully utilize the area, on the other words to produce zero unused space as shown in figure 1. The existence of many shapes can be assigned into area therefore, possible solutions to be analyzed will be huge. The analysis process has to be done because of every solution produces different number of items. Detail discussion of shape assignment problem can be referred at previous paper (Ismadi et al., 2010).

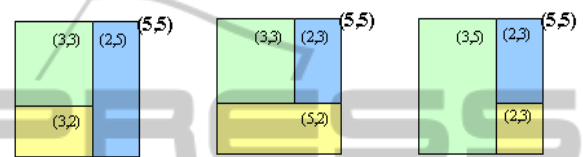


Figure 1: The three optimal solutions of (5,5) area coordinate

The same size of shape but different coordinate for instance (2,5) and (5,2) produce different number of items. Thus, the item calculation of combined shapes for every solution needs to be done. The maximum number of item is considered the best, so that the process of assigning shapes into an area will require a repetitive analysis. However, the existence of expected item number in the combined shapes is a wasted time to be analyzed. For instance, the first optimal solution as shown in figure 1 consists of the combined shapes that are represented by chromosome of 3, 3, 3, 2, 2, 5 of the 5, 5 area coordinate. This means the three combined shapes contain the three sequences of 3, 3; 3, 2 and 2, 5 coordinates, respectively. By changing the order of shapes, the combination of shapes can be 3,3; 2,5; 3,2 or 3,2; 3,3; 2,5 or 2,5; 3,3; 3,2 or ... or 2,5; 3,2; 3, 3. These all combined shapes certainly produce the same number of items. This situation should be avoided by ignoring the unwanted coordinate because it contributes to the increase of computation time in analysis process.

3 GENETIC ALGORITHM (GA) FOR SHAPE ASSIGNMENT PROBLEM

Shape assignment problem is considered as space allocation problem or packing problem in which

both are non-deterministic polynomial (NP) problems. Thus, an algorithmic intelligence techniques are required, so that the enormous decision arise can be answered in an acceptable time. The enormous increase in the number of decision has led to applying heuristic algorithms such as genetic algorithm (Stewart et al., 2004).

An analysis of clustering rectangles problem by Burke and Kendall (1999) showed the quality of the Simulated Annealing and Tabu Search algorithms better than the GA. However, another analysis in different problem domains such as TSPs (Pham and Karaboga, 2000), showed GA outperformed the others. Thus, the inconsistency of the efficiency and effectiveness of the algorithms give an indication that GA has space for improvement.

GA was applied to overcome shape assignment problem. However the huge iterations of analysis in GA as discussed in section 3.1 requires high computation time. Therefore, the two improvements of GA are identified in this issue towards reducing the number of iteration process and as a result diminish computation time.

3.1 Basic GA (BGA)

The common works of GA that consists of population initiation followed by the three operators of selection, crossover and mutation. The GA is typically able to solve some optimisations problems; however the computation time is always questionable. BGA refers to two situations as discussed below:

First situation refers the use of random number in assigning gene value. To find the number of possible solutions to be analyzed subjects to the number of shapes. With four shapes for example, means it requires eight genes of chromosome. Each gene is randomly assigned with value between 0 and 9 that is represented by X and Y coordinates respectively. The possible solutions in the worst case would be 100,000,000 ($10^8 = 10^{\text{chromosomeLength}}$).

Second situation devotes the existence of same value of sequence genes. The repetition of optimal shape solutions need to be compared with the purpose of finding the maximum number of item. However, same value of sequence genes will produce the same number of items and thus, the process of finding the optimal solution is a wasted time. According to Richard (2000), without controlling this situation usually to be reevaluated consequently spends computational effort on evaluating fitness function far exceeds that of genetic operator.

3.2 Improved GA (IGA)

The two situations in BGA for this domain problem therefore, we developed the two strategies with intention to reduce iteration process in analysis process and eventually the processing time will be probably reduced.

First strategy is to assign possible shapes into the area randomly with specific-domain in initialization process. This strategy focuses to randomly assign the value of genes that are derived from the X and Y coordinates of area as shown in figure 2. For example, the X and Y of area coordinate represent 4 and 5 respectively. Thus the odd spaces of chromosome the number can be assigned is between 1 and 4, similar to even spaces where the number between 1 and 5 is allowed.

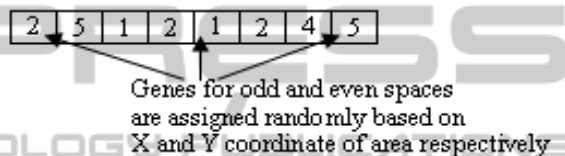


Figure 2: Chromosome by gene value that assigned randomly.

The possible solutions are based on the formula of $(X \text{ areaCoordinate}^{\text{chromosomeLength}/2} * Y \text{ areaCoordinate}^{\text{chromosomeLength}/2})$. Thus, the number of possible solutions by (4, 5) area coordinate is 160,000 ($4^4 * 5^4$) in the worst case.

Second strategy, the optimal shape assignment solution is not necessary the best solution as long as it does not achieve the maximum number of items. The process of obtaining the maximum item might need to be done iteratively. Therefore, the same optimal solution might occur. To avoid that, a control in GA by short term memory approach was employed. The short term memory is cleared at the start of each new generation and so can only store information about the current. The use of short term memory is to check whether an individual is a duplicate of one encountered earlier in the current generation (Jason and Chris, 2003).

Some cases the use of priori knowledge is applicable for feasible solution and protecting the same chromosome to be frequently reevaluated will probably reduces the computation time. There are two possible matters will occurs in generating optimal solutions which are 1) the current result and previous result is same 2) The block combination of current and previous result is same but different location place. These matters promote same number of items.

The creation of database is to collect all previous successful optimal results. The optimal solution will be compared with the available previous optimal solution in database. The existence of same combination of shapes will be rejected and regenerate to another optimal solution, otherwise the next process to calculate item and find the maximum item will be implemented. This comparison process is repeated until meeting a defined evaluation number. This control is expected reduce time because of a calculating of items for combined shapes can be skipped when the existence of same optimal between the current and previous solutions.

The flowchart in figure 3 shows the overall processes of finding the best solution of shape assignment.

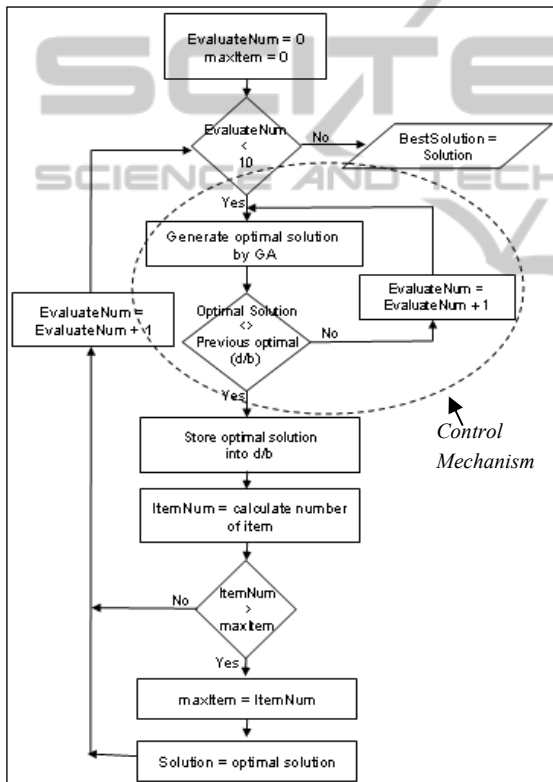


Figure 3: Processes of finding the best shape assignment.

4 RESULT AND DISCUSSION

The three experiments with different datasets were used for the comparisons between BGA and IGA in terms of computation time usage. The purpose of the first experiment is to find the number of chromosome to find optimal solution and the

number of iteration. Second experiment focuses on number of negative value and number of positive value that both were generated by BGA and IGA. The third experiment is to obtain the number of repetition when the execution of short term memory.

4.1 Experiment 1: Iteration of Genome to Produce the Successful Shape Assignment

Table 1 shows the same dataset was used in BGA and IGA. The chromosome value and gene iteration of both BGA and IGA were stated.

Table 1: Different Area Coordinate, Same Number of Shape.

Analyse Num	DATASET			BGA		IGA	
	Area Coordinate (x, y)	Number of Shapes	Size	Chromosome Value	Genome / Iteration Num	Chromosome Value	Genome / Iteration Num
1	4, 4	16	3	423212	725 / 2173	31341	131 / 393
2	5, 5	25	3	531243	1023 / 3069	52323	361 / 1033
3	6, 8	48	3	581216	2179 / 6537	373435	675 / 2025
4	8, 7	56	3	861171	2860 / 8569	573235	1858 / 5574
5	9, 9	81	3	594841	4835 / 14481	779227	2729 / 8187

The graph in figure 4 shows, both BGA and IGA generated a consistent increase of iteration number when the X and Y area coordinate raise. However, the iteration number of BGA is higher than IGA at all levels of the area coordinate; therefore the computation time taken by IGA less than BGA.

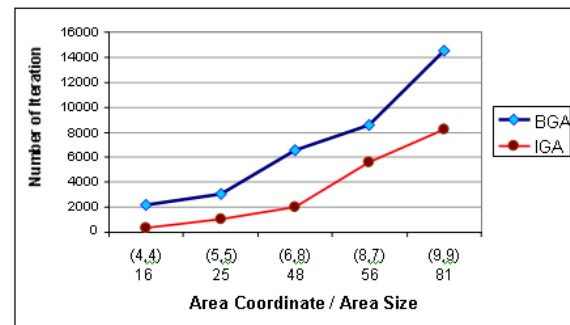


Figure 4: Iteration number of both BGA and IGA.

4.2 Experiment 2: Analysis of Positive and Negative Value of Fitness Status

The fitness status can be positive or negative value. The positive is given when some hard constraints have been fully fulfilled. While fail to do so the negative value will be assigned. The below procedure shows how the fitness status is given as follow:

$$\begin{aligned} \text{Area Size} &\rightarrow \text{areaX} * \text{areaY} \text{ ---- (a)} \\ \text{Size of combined shape} & \\ &\rightarrow \sum_{i=1}^N ((\text{shapeX})_i * (\text{shapeY})_i) \text{ ---- (b)} \\ \text{if (b) } &\leq \text{(a)} \\ &\text{fitness status} = \text{“positive”} \\ \text{Else if (b) } &> \text{(a)} \\ &\text{fitness status} = \text{“negative”} \end{aligned}$$

The positive values of chromosomes mean there are tendencies to assign all combined shapes into an area but the space is not necessarily fully utilized. However, at the next generation, these chromosomes have opportunity to reach optimal solution. Whereas the negative value of chromosomes are stated when a possibility of at least one of shapes fail to be assigned into the area. These chromosomes will be rejected for the next generation.

Table 2 shows the repetition number of positive and negative value for both BGA and IGA using the same dataset.

Table 2: Comparing possibility number of negative value in BGA and IGA.

Analysis Num	DATASET			BGA		IGA	
	Area Coordinate (x, y)	Number of Shapes	Chromosome value	Repetition number of positive value	Repetition number of negative value	Repetition number of positive value	Repetition number of negative value
1	4, 4	3	334113	103	254	306	39
2	5, 5	3	441541	205	265	217	16
3	6, 8	3	456325	2565	323	2399	197
4	8, 8	3	658325	3625	409	1301	111
5	9, 9	3	762693	616	75	494	38

The result of negative and positive value of both BGA and IGA were analyzed and the result is illustrated by graph as shown in figure 5. At 4, 4 and 5, 5 areas coordinate of BGA produce 70 and 57 percent of negative values respectively, and then the values tremendously plunge to 10 percent at the next coordinate of areas. While IGA shows the negative value of all areas is not much difference between 7 to 10 percent.

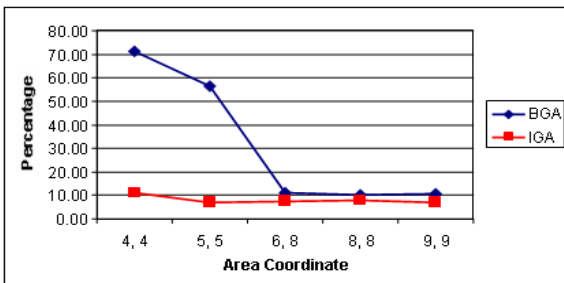


Figure 5: Percentage of repetition number of negative value.

The 4, 4 and 5, 5 area coordinate of BGA produced high percentage of negative value and it give a justification of the 0 to 9 gene values representation is not applicable. The higher gene values than area coordinates produce more unsatisfied conditions. We can conclude based on the overall result that the BGA produces more repetition number of negative value than IGA, so that insignificant chromosomes lead to time increase and affect to quality of result. Therefore, the undefined specific initialization in BGA becomes the possibility of chromosomes to violate the hard constraints are higher. These insignificant chromosomes have possibility to be used in processes of crossover and mutation. Therefore use of awful chromosome for producing new offspring consequently gives more time to find the optimal solution.

4.3 Experiment 3: Repetition Number of Same Solution Optimal Solution

Basically, the larger area coordinates will produce higher number of possible solutions. Refer to table 3, the area coordinates of analysis number 1 and 3 generated possible results with 4,096 ($4^3 * 4^3$) and 110,592 ($6^3 * 8^3$) respectively, as the result of analysis number 3 required more analysis time.

Refer to experiment 3, there were three same solutions as shown in table 3. As a comparison, by BGA, the process of calculating tree number and determining the best line-direction and required $15,575 (\sum_{bs=1}^{bs=N} (ItNum * OS)_{bs})$ iterations, while IGA required only $9203 (\sum_{bs=1}^{bs=N} ItNum_{bs})$ iterations.

The overall time taken is based on the accumulation of time in all repetition processes. Small areas coordinates have higher tendency to occurrence of the repetitive optimal shape solution. The different computation time of both BGA and

Table 3: Analysis of the Same Optimal Shape Solution.

Optimal Block Solution (bs)	Shape Coordinates (Genes)	Iteration Number of Item (ItNum)	Repetition of Same Optimal Solution (OS)
1	341113	2842	1
2	333114	3186	3
3	331341	3175	1

IGA increases consistently when more repetition number of same optimal solution. However, when

the evaluation number is small (refer to figure 3), the possibility of repetition number might not happen, thus computation time of both will be same.

From our observation the taken time for each experiment was inconsistent at certain time but it infrequently happen is a challenge. A justification on this matter is a probabilistic algorithm with a randomness strategy applied in GA, therefore the number of repetitions and iterations and hardly expected.

5 CONCLUSIONS

IGA possibly reduces number of repetition by focussing on assigning values to genes and controlling the repetition of optimal solution. The gene value is based on an area coordinate will be more significant when the area coordinate increases. Besides that, the less number of negative values in obtaining the optimal solution will reduce computation time because of the awful chromosomes will be diminished. Meanwhile, controlling mechanism in obtaining the best optimal reduce computation time by looking at the number of iteration.

ACKNOWLEDGEMENTS

This research is registered in the Fundamental Research Grant Scheme (FRGS) and it is fully funded by Ministry of Higher Education (MOHE), Malaysia. Authors express our highly appreciation and thanks to MOHE, Malaysia and Universiti Teknologi MARA, Malaysia for sponsoring one of the authors in PhD level. Authors also wish to thank the Universiti Putra Malaysia that provides facilities and conducive environments for carrying out this research.

REFERENCES

- Blum, C., Roli, A., 2003. Metaheuristic in Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Journal*, 268-238.
- Burke, E., Kendall, G. 1999. Comparison of Meta-Heuristic Algorithm for Clustering Rectangle. *Computer and Industrial Engineering* 37, 383-386.
- Chen-Fang Tsai, Kuo-Ming Chao, 2007. An Effective Chromosome Representation for Optimising Product Quality, *Proceedings of the 2007 11th International*

- Conference on Computer Supported Cooperative Work in Design*, 1032-1037.
- Eiben, A. E., Schippers, C. A., 1998. On Evolutionary Exploration and Exploitation, *Fund. Inf.* 35, 1-16.
- Ismadi, M. B., Abu Bakar, M.S., Md Nasir, S., Ali, M., Mahmud, T.M.M., 2010. Shape Assignment by Genetic Algorithm towards Designing Optimal Areas, *International Journal of Computer Science Issues IJCSI, Vol. 7, Issue 4, No 5*, 1-7.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.
- Jason, C., Chris, H., 2003. Improving Genetic Algorithms Efficiency Using Intelligent Fitness Functions, *IEA/AIE 2003, LNAI 2718*, pp. 636-643.
- Miihlenbein, H., Schlierkamp, V.D., 1993. Predictive Models for the Breeder Genetic Algorithm I. *Evolutionary Computation*. 1(1), 25-50.
- Pham, D. T., Karaboga, D., 2000. *Intelligent Optimisation Techniques: Genetic Algorithm, Tabu Search, Simulated Annealing and Neural Network*, Springer-Verlag.
- Rae-Dong Kim, Ok-Chul Lung, Hyochoong Bang, 2007. A Computational Approach to Reduce the Revisit Time Using a Genetic Algorithm, *International Conference on Control, Automation and Systems* 2007, Oct. 17-20, 2007 in COEX, Seoul, Korea, 184-189.
- Richard, J. P., 2000. Comparing Genetic Algorithms Computational Performance Improvement Techniques, *Artificial Neural Networks in Engineering, Proceedings*, 305-310.
- Schaffer, J. D., 1985. Learning Multiclass Pattern Discrimination., *Proceedings of the First International Conference on Genetic Algorithms*, 74-79
- Srinivas, M., Patnaik, L., 1994. Genetic Algorithms: A Survey. *Computer. IEEE Press*, 17-26.
- Stewart, T. J., Janssen, R. and Herwijnen, M. V. 2004. A Genetic Algorithm Approach to Multiobjective Landuse Planning, *Computers & Operations Research Journal*, 2293-2313.
- Stutzle, 1999. Local Search Algorithms for Combinatorial Problem – Analysis, Algorithms and New Applications. *FISKI – Dissertationen zur Kunstliken Intelligenz. Infix, Sankt Augustin, Germany*.