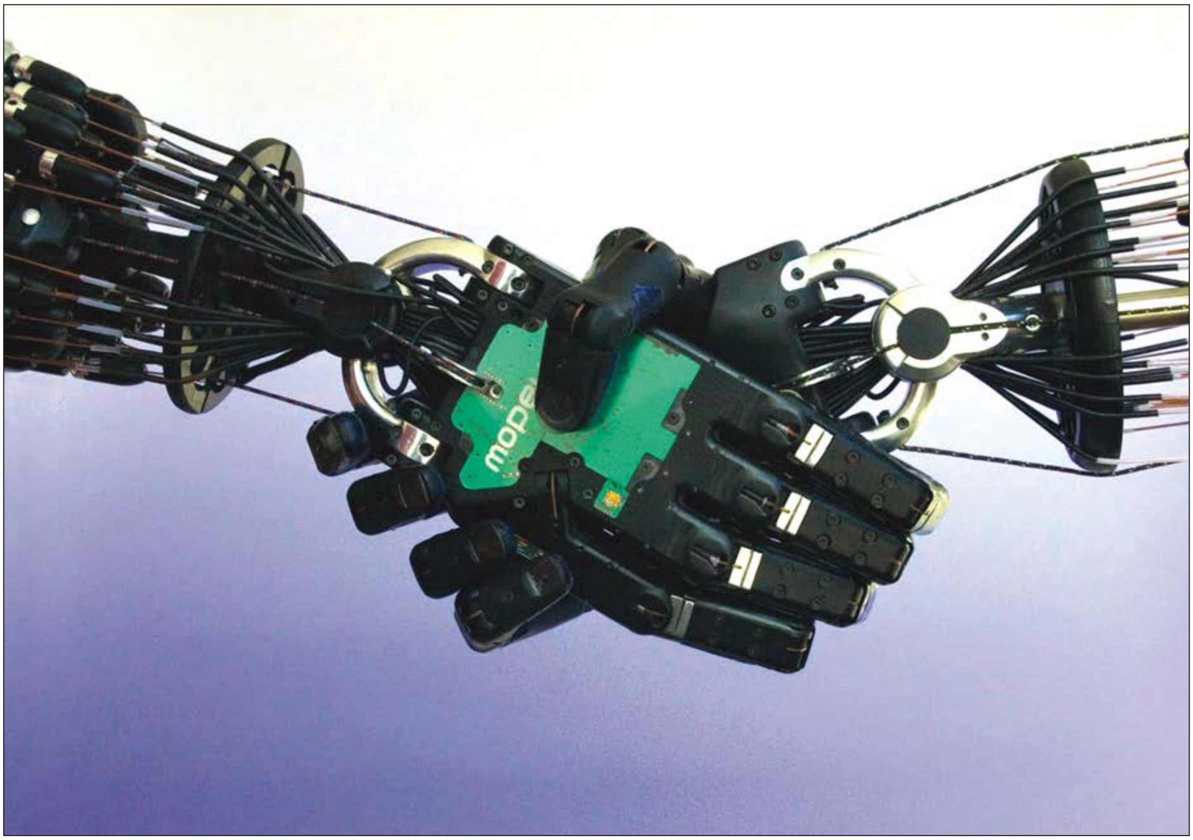


Journal of Artificial Intelligence

ISSN 1994-5450

Journal of Artificial Intelligence 5 (2): 64-75, 2012
ISSN 1994-5450 / DOI: 10.3923/jai.2012.64.75

A Framework of Genetic Algorithm Improvement for Optimal Block Division in Lining Layout Planning

[1]I.M. Badarudin, [2]A.B.M. Sultan, [2]M.N. Sulaiman, [2]A. Mamat and [3]M.T.M. Mohamed
[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia
[2]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia
[3]Faculty of Crop Science, Universiti Putra Malaysia, Malaysia

Corresponding Author: I.M. Badarudin, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia

ABSTRACT

This study focuses on the Genetic Algorithm (GA) as solution strategy for block division in Lining Layout Planning (LLP). Block division is optimal when the combined shapes in an area promote no empty space. This problem requires huge number of possible solutions to be analyzed and it is considered as a set of space allocation problems. Classical Genetic Algorithm (CGA) with basic operators was applied to find optimal solution. Despite CGA is able to promote the optimal result however it has opportunity to improve time efficiency. Therefore, a framework of GA improvement (IGA) for block division was introduced by looking into the genes of chromosome for problem representation and prior to the processes of crossover and mutation. The IGA involves three strategies which are; (1) specific random value for chromosome representation, (2) deterministic crossover is to avoid from the same result of crossover process and (3) deterministic mutation is to protect overlapping shapes. This paper reported the theoretical analysis of possible improvements and then generates results from the various coordinates of areas to evaluate the performance of the CGA and IGA. The overall result presents that IGA promoted fewer number of repetitions than CGA and as a result IGA expedites processing time to obtain optimal result.

Key words: Genetic algorithm, specific random value, deterministic crossover, deterministic mutation, block division

INTRODUCTION

The main drawback of a Genetic Algorithm (GA), especially when applied to large instances of complex design problems, is perhaps an excessively long run-time that is needed to deliver some satisfactory results (Lech and Adam, 2002). Every generated chromosome in initial generation through the analysis process to accomplish the available constraints and conditions require amount of time depends on the problems complexity. Therefore, the strategy to improve the efficiency of GA focusing in time is a significant contribution in this domain issue. Population initialization, crossover and mutation are important properties toward successfully finding an optimal solution in GA. The random strategies with probabilistic solutions in GA make the computational time are difficult to expect.

Population initialization plays a main rule on the convergence speed solution quality of genetic algorithm. Thus, some works on population initialization have been deployed to improve GA performance. According to Xiaozhen *et al.* (2008), a simple method to generate the initialization population is random generating. By random technique the number is generated randomly to produce solution. This method is easy to be performed but can not assure the quality of initialization population. Other technique to initialize population is by setting approximately known solutions when we have a priori knowledge about given optimization problem. This technique, however, imposes some bias in terms of how it looks for better solution and where it looks in the search space. Langer scale of initialization population means the GA's search space is unlimited and as a consequence the possibility of trapping at local optimal (the algorithm stops in certain part that eventually fail to achieve the global solution) is low but it may increases computation time will eventually decrease the performance of GA. An inadequate scale of community will influence on comprehensive performance of GA (Hopper and Turton, 1997). 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 than using a small population size (Grefenstette, 1986; Schaffer *et al.*, 1989).

The crossover and mutation operators are a pulse in successful Genetic Algorithm (GA). The crossover acts through the genes exchange of two parents, while, mutation is a slight change of selected genes in which the both processes produce the new offspring in the next generation. According to the published papers, researchers have dedicated a special intention to crossover, while the mutation has always been seen as a secondary operator (Goldberg, 1989; Holland, 1975). Despite the number of research on crossover are much higher than mutation but both share a same inspiration towards avoiding from falling into premature convergence or getting trap at local optimal. The improvement of these properties by deterministic GA will probably give a significant effect to diminish time. Some proposed techniques in deterministic population involve genetic algorithm operators (Lech and Adam, 2002; Laredo *et al.*, 2009; Salomon, 1999; Ponslet *et al.*, 1995), deterministic crossover (Cheng *et al.*, 2010; Pasi, 2000), deterministic mutation (Kanarachos *et al.*, 2003; Fukumi and Akamatsu, 1998) were organized since there are spaces to make an improvement of GA performance in solution analysis. Other improvements by adaptive GA were also proposed (Ismail and Irhamah, 2008; Taishan, 2010; Ye *et al.*, 2010).

Algorithm complexity is measured in terms of time complexity and space complexity. Although, now-a-day, computers are very fast but they are not infinitely fast and computer memory may be cheap but it is not free. Computing time is therefore, a bounded resource and so is space in memory (Thomos, 2007). These resources should be used wisely, thus this is a challenge for computer scientist to manage algorithms efficiently in terms of time or space. The efficiency of an algorithm is accessed in terms of the computing resources that are needed to execute the algorithm and this includes execution time and space. Researchers are paying more attention in minimizing time complexity of algorithm because there are spaces for improvement focusing on the algorithmic strategies in general or specific problems. Moreover, in theory the efficiency of algorithm is usually addressed to its time complexity. While less attention of space because of the computer memory costs have decreased over the past 25 years, thus the improvement is not a significant contribution. To determine the efficient algorithms refers to speed taken by an algorithm to produce results. The time complexity of an algorithm is an amount of time required to execute an algorithm for a given number of input (Ruibin, 2005). The execution time refers to the number of repetitions that algorithm takes to process the input and produce a solution. It is measured by its rate of growth relative to standard function such as constant, logarithmic, polynomial and exponential.

This study devotes to an improvement from classical GA (CGA) by focusing on the genes for problem representation and for feasible chromosomes prior to the process of crossover and mutation;

this led to the proposed strategies of specific random value, deterministic crossover and deterministic mutation. The theoretical discussion shows there is a high tendency for enhancing the GA efficiency, hence it has motivated us to propose a framework of GA improvement (IGA) with detail discussion of its strategies and then to conduct an intensive analysis to see the performance of IGA. Therefore, the ultimate objective of this research was to find the computational time processed by CGA and IGA by looking into the number of iteration and processing time.

OVERVIEW OF BLOCK DIVISION COMPLEXITY

In lining layout planning (LLP), optimal design with the strategies of determining block division and then choosing the best lining 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. To decide the optimal combined shapes has huge number of possible solutions as a result might promote ambiguous solution.

The optimization tree plantation areas by LLP technique is a complex solution. Therefore combining of Integer Programming (IP) and GA towards determining the highest number of trees was applied (Ismadi *et al.*, 2010a). In block division, we attempt to find optimal solution whenever the combination of shapes must completely be fitted with no unused spaces. The existence of unused space at least one shape beyond the border of area will be rejected. Block division by shape assignment strategy has been discussed in the previous paper (Ismadi *et al.*, 2010b) in which focuses the strategy to assign the determined number of shapes into an area. Therefore, to find the combination of the determined number of shapes in an area has two constraints. First, combined shapes according to the determined block number must be fully utilized the area with zero unused space. Second, prevents possible shapes from overlapping. The overlapping shapes share same planting direction, so that the shapes have possible to be merged will eventually fail to find the required number of blocks.

The possible solutions to be analyzed rely on the number of blocks (BlockNum) in an area or the number of shapes (ShpNum) that represent blocks. For example, the (4, 5) coordinates of area 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 Fig. 1. W and H coordinated of area derived from x4 and y4 respectively.

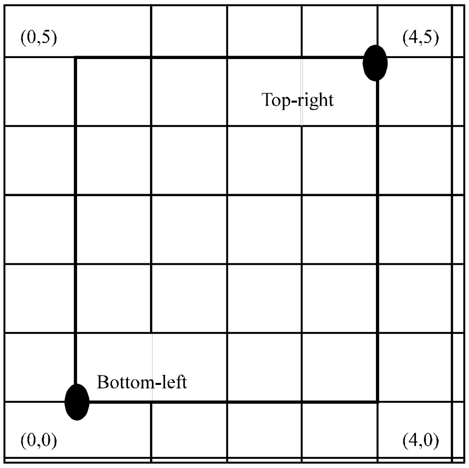


Fig. 1: Representation of (4, 5) area coordinate

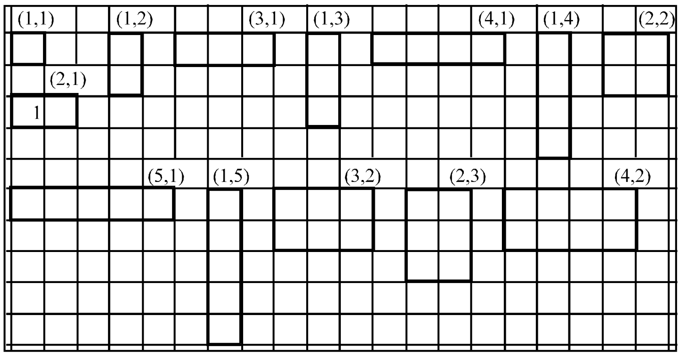


Fig. 2: Possible shapes to be used refers to the area coordinate of (4, 5)

The possible shapes can be (1, 1), (1, 2), (1, 3), (1, 4), ..., (4, 3), (4, 4), (4, 5) which is the number of shapes can be used is 20 (W*H) as shown in Fig. 2. To find optimal solution can be simplified by ($\text{ShpNum}^{\text{BlockNum}}$). Let area coordinate is (4, 5) thus, ShpNum = 20, BlockNum is 5, the iteration number for possible solutions in the worst case for block solution will be 3.2 (20^5) million iterations.

Therefore, more ShpNum or BlockNum will increase number of iterations and eventually rise time for solution. Therefore, the processing time is exponential and this is considered as non-deterministic polynomial (NP) problem. According to Thomos (2007), this is considered as NP in which it is a class of problems that are verifiable in polynomial time that we can verify $O(n^k)$ time whenever a proposed solution is correct.

CLASSICAL GENETIC ALGORITHMS (CGA)

The CGA employed basic works of population initialization and the three operators such as selection, crossover and mutation. The genes value of a chromosome is based on probabilistic with randomness strategies. A chromosome has a number of genes. Since a shape represented by the pairs of odd and even spaces, the genes number based on Block Num*2. Genome is part of solutions in a chromosome, thus a pair of shape is considered as a genome. The value of genes relies on the process of assigned the random number as shown in Fig. 3.

The process of shape assignment begins by assigning random values to the genes of a chromosome that represents an individual. The chromosomes consists of shapes will be assigned into an area. The search process guided by the first two genes of chromosome which is considered as first shape from the Bottom-Left as shown in Fig. 4. Then, to fill in the others shapes are evaluated by strategies of Bottom-Left by Right first (BL-Rf) and Bottom-Left by Top first (BL-Tf). 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.

The CGA shows the satisfactory solution towards successfully achieving global premature convergence especially using appropriate population initialization however the time taken is always questionable. The unspecific random value in CGA based on value between 0 and 9 to be assigned randomly into genes in which the number of possible solutions in the worst situation will be $10^{\text{chromosome length}}$. This situation leads to higher computation time when involving huge number of genes. Moreover, assigning genes value by unspecific random value in CGA leads to the insignificant values which reduce feasible chromosomes at initial space. Thus, possibility to achieve

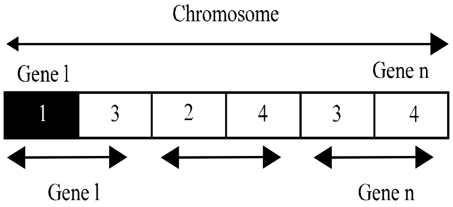
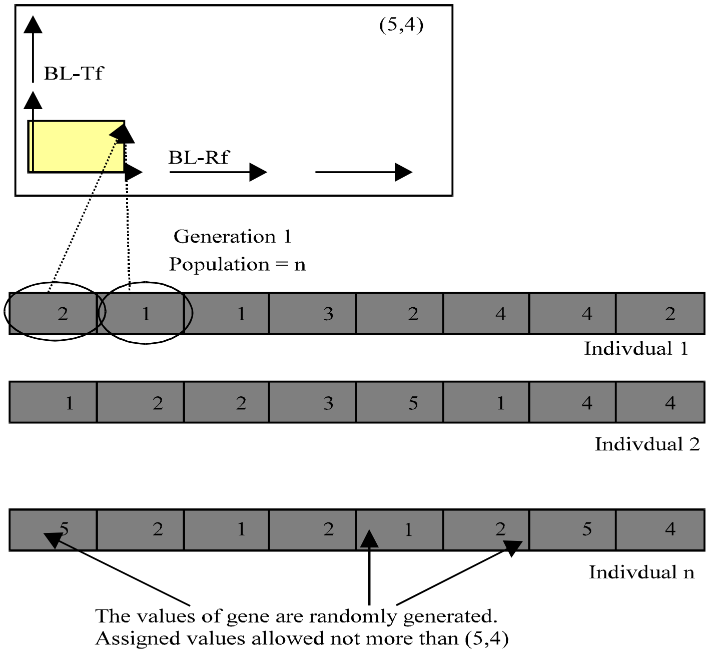


Fig. 3: Illustration of a chromosome



The values of gene are randomly generated.
Assigned values allowed not more than (5,4)

Fig. 4: Shape assignment strategy

optimal result is low and requires more number of generations for other solutions leads to spend much time. While, in crossover or mutation the infeasible chromosomes that can be expected have no significant results is a wasted time to be analyzed will influence the length of time.

FRAMEWORK OF GA IMPROVEMENT

The strategies involve specific random value, deterministic crossover and deterministic mutation. Therefore, the processing time in GA can perhaps be diminished when the expected unnecessary analysis for infeasible chromosomes are excluded at the initial operation of stages.

The framework of GA improvement in Fig. 5 illustrates analysis process for possible solutions at stages of GA's selection and its operators to generate an optimal block solution. Fail to reach the optimal solution in current generation; the new offspring will be produced for the next generation. The chromosomes of the generation will again, be analyzed. This process is terminated whenever the optimal solution obtained, otherwise selection process evaluate the chromosomes' fitness for seeking the feasible chromosomes for the next stages of crossover and mutation.

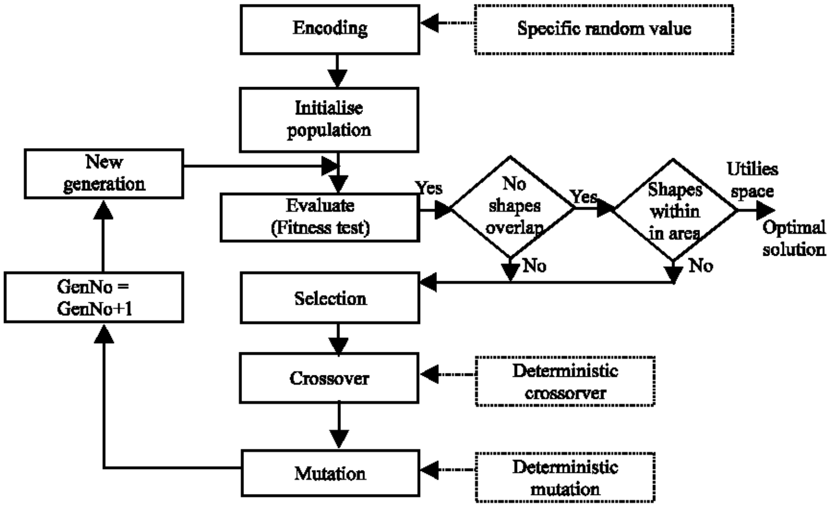


Fig. 5: Framework of GA improvement

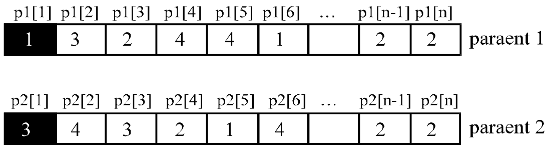
Encoding process by strategy of specific random value is to randomly encode the genes values of chromosomes according to a given area coordinate. This strategy is to reduce possible solutions and has potential to obtain optimal solution in lesser time.

Focusing on, to efficiently explore the search space refers to the strategies of deterministic crossover and mutation. Both strategies are to monitor the gene values in a chromosome for selecting the feasible chromosomes before applying the process of crossover or mutation. The overlapping shapes of solution means a chromosome promotes same gene values of its pairs is considered as infeasible, this chromosome is recommended to be skip for manipulation process in crossover or mutation. This deterministic technique that applied into crossover and mutation will probably reduce the infeasible offspring.

Specific random value: Specific random value refers the genes values of a chromosome depends on the W or H of an area coordinate. The generated number of possible solutions based on the formula of $W^{odd\ length}\times H^{Even\ length.}$ Let W and H of area represented by 4 and 5, respectively (Fig. 1), thus each odd space of chromosome can be assigned between 1 and 4 numbers, similar to even spaces where a number between 1 and 5 for each is allowed. Thus, the number of possible solutions by (4, 5) area coordinate is 160,000 ($4^4\times5^4$) which reduce from 100,000,000 (10^8) by unspecific random value as discussed earlier. This strategy reduces the number of iteration process for possible solution and eventually improves time efficiency.

Less number of possible solutions maintains diversity for exploration and avoids from getting trapped at local optimal and perhaps obtains a fast global convergence. Moreover, specific random value rejects the infeasible chromosomes that violate the basic constraints to give the algorithm long enough run-time for searching an optimal solution therefore the algorithm may suggest optimal result faster.

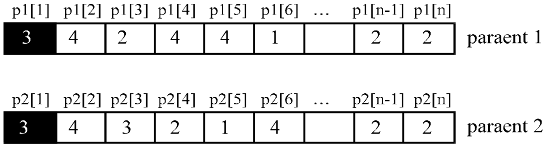
Deterministic crossover: In deterministic crossover strategy is to priory determine the result of crossover exchanged gene values of two parents that yield the same value of chromosome as a consequence no significant effect to new produced offspring. The Fig. 6 and 7 show the new



p1[1]	p1[2]	p1[3]	p1[4]	p1[5]	p1[6]	...	p1[n-1]	p1[n]	
1	3	2	4	4	1		2	2	paraent 1

p2[1]	p2[2]	p2[3]	p2[4]	p2[5]	p2[6]	...	p2[n-1]	p2[n]	
3	4	3	2	1	4		2	2	paraent 2

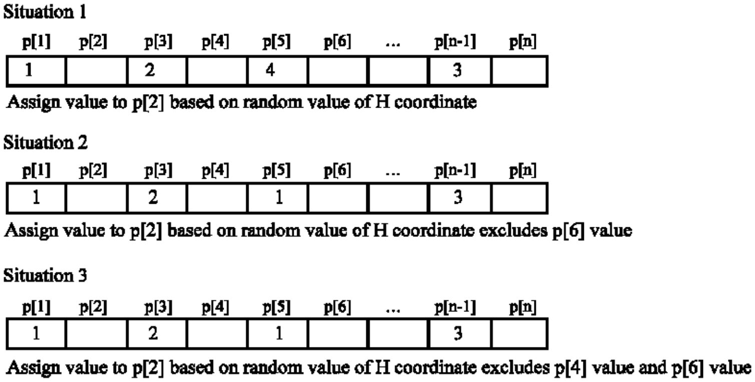
Fig. 6: Crossover process of two parents is accepted



p1[1]	p1[2]	p1[3]	p1[4]	p1[5]	p1[6]	...	p1[n-1]	p1[n]	
3	4	2	4	4	1		2	2	paraent 1

p2[1]	p2[2]	p2[3]	p2[4]	p2[5]	p2[6]	...	p2[n-1]	p2[n]	
3	4	3	2	1	4		2	2	paraent 2

Fig. 7: Crossover process of two parents is rejected



Situation 1

p[1]	p[2]	p[3]	p[4]	p[5]	p[6]	...	p[n-1]	p[n]
1		2		4			3	

Assign value to p[2] based on random value of H coordinate

Situation 2

p[1]	p[2]	p[3]	p[4]	p[5]	p[6]	...	p[n-1]	p[n]
1		2		1			3	

Assign value to p[2] based on random value of H coordinate excludes p[6] value

Situation 3

p[1]	p[2]	p[3]	p[4]	p[5]	p[6]	...	p[n-1]	p[n]
1		2		1			3	

Assign value to p[2] based on random value of H coordinate excludes p[4] value and p[6] value

Fig. 8: The situations of assigning values in mutation process

produced offspring after swapping a pair (the pair location for swapping based on random strategy) of the two parents. The process of crossover in Fig. 6 is accepted because of both parents contribute different values of the selected pair. Figure 7 in contrast the same pair as expected promotes insignificant new offspring, thus the process of swapping will be ignored.

Number of insignificant chromosome based on the formula of n×(W×H), where n is number of shape used. A shape has possible to promote W×H possible solutions. Let H = 9 and W = 9 and chromosome length is 8 that means n is 4 (chromosome length/2). In the worst case, 324 (4 (9×9)) infeasible chromosome will be skipped for the swapping process in crossover.

Deterministic mutation: As the constraint of no overlapping, the value of genes for each pair must be different each other. Therefore, the deterministic mutation aims to determine the possibility of producing the similar genes when the new offspring are produced.

In deterministic mutation strategy, we randomly choose a pair of a chromosome. We show the selected odd or even space of a pair for mutation process will be excluded if there is a possibility to produces same pair value with others spaces as depicted in Fig. 8. Let us say, the mutation process occurs at p[2]. The value between 1 and value of H area will be randomly assigned into p[2] as shown in situation 1, however, some values cannot be used for situation 2 and 3. Situation 2, same value of p[1] and p[5] causes the random value to be placed is exclusive of p[6]. For situation 3, the value of p[4] and p[6] will be skipped whenever p[3] and p[5] share same value with p[2].

EMPIRICAL ANALYSIS

Analysis design: The objective was to compare the effectiveness of CGA and IGA. We used the same parameters of population size, crossover and mutation rates for both to avoid bias of results. Refer to 100 populations, 75 crossover rates and 5 mutation rates that are based the standard or accepted parameters. The analysis consists of three parts as follow:

- Comparison between the result of specific random value and unspecific random value. The six types of different random values were used to be placed in 4, 4 coordinate of area. We stated the number of positive and negative fitness value from the stating crossover (when start generate) and final crossover (when meet optimal solution), number of generation and taken time to obtain optimal solution. Then, with different areas coordinates, we stated the number of positive and negative value produced by both strategies
- Comparison between basic crossover and deterministic crossover where five coordinates of areas were selected. We stated the number of generation and taken time to obtain optimal solution of both
- Comparison between basic mutation and deterministic mutation in which five coordinates of areas were selected. We stated the number of generation, taken time to obtain optimal solution and average skip number of both

RESULTS AND DISCUSSION

Specific random value: This analysis focuses on an area with 4, 4 coordinate in which the use of random value for x (odd spaces) and y (even space) starting from 4 and 4, 5 and 5 until 9 and 9, respectively as shown in Table 1. One of the reasons, a result fails to find optimal solution because size of combined shapes is larger than area. While the smaller size means the chromosomes have tendency to be regenerated in crossover or mutation for other solutions. Therefore, we set if a chromosome has size of combined shapes less than or equal to area size (assigned Shapes Size≤Area Size), thus the chromosome is assigned by a positive values, otherwise is negative value. The percentages of both values at initial crossover and final crossover were stated together with the chromosome, generation number and time that reach optimal solution.

Result shows the random value of 4, 4 promotes the smallest number of generation and fastest time than other values to find optimal solution. The 7, 7 until 9, 9 however fail to find optimal solution after 100th generations. The result of higher positive and less negative number is better because to find the optimal solution is faster, hence 4, 4 is the best case. From this result we found that the increase of x and y values, will decrease positive value for initial crossover and final crossover; number of generation will increase to meet optimal solution as a result time taken will also rise. The reason of 7, 7; 8, 8 and 9, 9 fails to find optimal solution because of the total size of combined shapes is larger than area.

From the above discussion the random value that refers to area coordinate W, H is best preference. This is because of the possibility of shapes to exceed W, H is less, so that time to achieve optimal is faster. Coordinate x or y more than W or H requires more possible solutions to be analyzed, besides that the possibility to chromosomes exceeds H or W values which means these chromosomes do not promote to optimal solution is higher. These factors consume more time to obtain optimal solution. As expected the random x = 9 and y = 9 has highest possibility to exceed W, H consequently requires much more time to generate the result.

Table 1: Results of different random values for 4, 4 area coordinate

Random	Initial crossover process		Final crossover process				
value for	+ve (%)	-ve (%)	+ve (%)	-ve (%)	Optimal solution	Generation No.	Time taken (sec)
4, 4	83	17	87.27	12.73	333114	3	0.54
5, 5	52	48	61.32	38.68	334113	9	1.24
6, 6	30	70	42.15	57.85	131131	31	4.34
7, 7	16	84	20	80	Fail to find optimal solution after 100th generation (27.12 sec)		
8, 8	16	84	16	84	Fail to find optimal solution after 100th generation (27.46 sec)		
9, 9	7	93	7	93	Fail to find optimal solution after 100th generation (27.54 sec)		

Table 2: Comparison between specific and unspecific random value

Dataset				Unspecific random value		Specific random value	
Analysis No.	Area coordinate (W, H)	No. of shape	Chromosome value	No. of positive value	No. of negative value	No. of positive value	No. of negative value
1	4, 4	3	341113	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

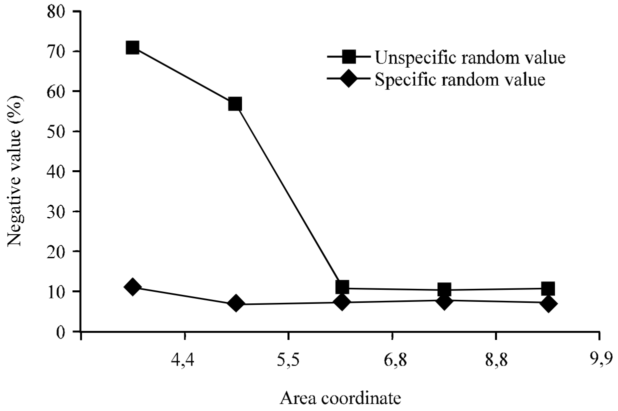


Fig. 9: Percentage of negative by specific and unspecific random value

We extended the analysis by various areas coordinates using specific and unspecific random value to generate optimal solution for combined three shapes. The positive and negative number promoted by specific and unspecific random value based on the condition of (assigned shapes size≤area size) as the above discussion was stated as shown in Table 2 for comparison purposes.

Figure 9 shows 4, 4 and 5, 5 coordinates of areas using unspecific random value produced high percentage of negative value of 71.15 and 56.38, respectively and others are level between 10 and 11%. While for areas coordinate of specific random value promoted the percentage of negative values in range from 6.87 to 11.30. Overall unspecific random value produced more repetition number of negative value than specific random value to find optimal solution.

Based on the both discussions show, the values to be assigned into genes is less or equal to W and H of coordinate has possibility to reach optimal is faster. We conclude

Table 3: Results of basic crossover and deterministic crossover

	Find an optimal solution			
	Basic crossover		Deterministic crossover	
Coordinate area (W, H)	Generation No.	Time taken (sec)	Generation No.	Time taken (sec)
4,4	5	0.084	4	0.066
5,5	9	0.128	8	0.102
6,6	9	0.134	9	0.108
7,7	16	0.234	14	0.156
8,8	71	0.983	30	0.306

Table 4: Comparison of time taken basic mutation and deterministic mutation

Area coordinate (W, H)	No. of generation	B-Mutation time (sec)	D-Mutation time (sec)	Average skip No.
4,4	3	0.054	0.050	23.33
6,6	4	0.074	0.070	26.50
7,6	12	0.176	0.170	26.67
7,8	6	0.100	0.096	28.44
9,9	7	0.104	0.096	19.71

that by the strategy of specific random value spend fewer time analyses because of the potential to produce infeasible chromosome is less.

Deterministic crossover: The analysis focuses on W and H coordinate of areas, the number of generation and computational time to reach optimal solution by basic crossover and deterministic crossover was stated as shown in Table 3.

The time usage by deterministic crossover and basic crossover time increased when the value of W or H raise. The overall shows the deterministic crossover spent less time in analysis process than basic crossover. The 8, 8 coordinates of area promotes a significant time difference where basic crossover higher by 0.667 (0.983-0.306) sec. Other coordinate areas also shows basic crossover higher with difference between 0.018 (0.084-0.066) sec and 0.078 (0.234-0.156) sec. This situation occurs when more number of infeasible chromosomes in basic crossover leads to much time to be analyzed. The generation number also increases according to the increment of W or H, however deterministic crossover produced less generation number than basic crossover for all coordinates of areas.

As conclusion that deterministic crossover with strategy focusing on the potential chromosomes for reaching optimal, therefore the rejection of infeasible chromosomes will reduce analysis number as a result time will be more efficient.

Deterministic mutation: The five areas with W, H coordinates were used. The number of generations and processing time for basic mutation and deterministic mutation were stated based on the success of reaching optimal solution in each area. The average skip number of each generation by deterministic was also reported. The processing time taken by both strategies are not consistent where sometime less areas size consume more time, this is because of GA is probabilistic algorithm.

The result shows larger generation number increased amount of time but the higher W, H does not necessarily spend more time (Table 4). The important finding is the time taken by basic mutation more than deterministic mutation for all areas. A conclusion of this result indicates the deterministic mutation by strategy to reject infeasible chromosomes before mutation or in another word focuses on the potential chromosomes for reaching optimal makes the process is more efficient.

CONCLUSION AND FUTURE WORKS

The IGA reduced number of repetitions by focusing on assigning values to genes and monitoring genes value for crossover and mutation process. As overall, combine the three strategies it will be more significant effect of reducing repetition number. The improvement of GA is important for the overall result in LLP because a series number of analysis by poor strategy may consume much processing time. The IGA improve time efficiency in block division solution. It will be more significant effect of reducing repetition number when the use of block or area coordinates increases. However, GA employs the probabilistic with random strategy, hence, some situations but infrequently occur the result of CGA shows better performance. It is a challenge to make sure the results is more consistent.

The code representation of genes uses the W and H coordinates of top-right area, therefore the odd and even pair of a chromosome is able to determine the size and pattern of a shape. This code representation is appropriate for rectangular area. To extend this study focusing on irregular areas, the coordinates of points need to completely collected and code representation might be different. Meanwhile, the deterministic crossover strategy in this study is to skip the process of the same pair of same pair of two parents to be interchanged. While deterministic mutation is to skip the process of the overlapping shapes. We assume that time will be more efficient if the deterministic crossover takes the consideration of overlapping shapes.

ACKNOWLEDGMENTS

This research is registered under the Fundamental Research Grant Scheme (FRGS). The grant number is FRGS/03-04-10-873FR/F1 and is fully funded by the Ministry of Higher Education (MOHE), Malaysia. The authors would like to express their highest appreciation and thanks to MOHE, Malaysia and Universiti Teknologi MARA (UiTM), Malaysia, for sponsoring one of the authors to pursue a Ph.D. The authors also wish to thank the Universiti Putra Malaysia (UPM) for providing the excellent facilities such as computer laboratory and required software as well as the conducive environments for discussion towards conducting this research.

REFERENCES

Cheng, C., K. Fallahi, H. Leung and C.K. Tse, 2010. An AUVs path planner using genetic algorithms with a deterministic crossover operator. Proceedings of the IEEE International Conference on Robotics and Automation, May 3-7, 2010, Anchorage, AK., USA., pp: 2995-3000.

Fukumi, M. and N. Akamatsu, 1998. Rule extraction from neural networks trained using evolutionary algorithm with deterministic mutation. Proceedings of the IEEE International Joint Conference on World Congress, Computational Intelligence, Neural Networks, May 4-8, 1998, Anchorage, AK., USA., pp: 686-689.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimisation and Machine Learning. Addison Wesley, Reading, MA.

Grefenstette, J.J., 1986. Optimisation of control parameter for genetic algorithm. Proceedings of the IEEE Transaction on System, Man and Cybernetics, January 1986, USA., pp: 122-128.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. 1st Edn., University of Michigan Press, Ann Arbor, Michigan, ISBN: 0472084607.

Hopper, E. and B. Turton, 1997. Application of genetic algorithms to packing problems: A review. Proceeding of the 2nd On-Line World Conference on Soft Computing in Engineering Design and Manufacturing, June 23-27, 1997, Springer Verlag, London, pp: 279-288.

Ismadi, M.B., A.B.M. Sultan, S.M. Nasir, M. Ali and T.M.M. Mahmud, 2010a. Novel genetic algorithm towards implementing a lining-layout optimization strategy. Int. J. Comput. Sci., 7: 318-326.

Ismadi, M.B., A.B.M. Sultan, S.M. Nasir, M. Ali and T.M.M. Mahmud, 2010b. Shape assignment by genetic algorithm towards designing optimal areas. Int. J. Comput. Sci., 7: 1-7.

Ismail, Z. and Irhamah, 2008. Adaptive permutation-based genetic algorithm for solving VRP with stochastic demands. J. Applied Sci., 8: 3228-3234.

Kanarachos, A., D. Koulocheris and H. Vrazopoulos, 2003. Evolutionary algorithms with deterministic mutation operators used for the optimization of the trajectory of a four-bar mechanism. Math. Comput. Simul., 63: 483-492.

Laredo, J.L.J., C. Fernandes, J.J. Merelo and C. Gagne, 2009. Improving genetic algorithms performance via deterministic population. Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, July 8-12, 2009, ACM, New York, pp: 819-826.

Lech, J. and P. Adam, 2002. Genetic engineering verses natural evolution, genetic algorithm with deterministic operators. J. Syst. Archit., 48: 99-112.

Pasi, F., 2000. Genetic algorithm with deterministic crossover for vector quantization. Pattern Recognit. Lett., 21: 61-68.

Ponslet, E., G. Maglaras, R.T. Haftka, E. Nikolaidis and H.H. Cudney, 1995. Structural Optimization. Vol. 10, Springer-Verlag, Berlin, pp: 247-257.

Ruibin, B., 2005. An investigation of Novel Approaches for Optimising Retail Shelf Space Allocation. Ph.D. Thesis, The University of Nottingham, UK.

Salomon, R., 1999. The deterministic genetic algorithm: Implementation details and some results. Proceedings of the 1999 Congress on Evolutionary Computation, July 6-9, 1999, Washington, DC., USA., pp: 695-702.

Schaffer, J.D., R.A. Caruana, L.J. Eshelman and R. Das, 1989. A study of control parameters affecting on-line performance of genetic algorithm for function optimization. Proceedings of the International Conference on Genetic Algorithm and their Application, June 1989, George Mason University, Washington, USA., pp: 51-61.

Taishan, Y., 2010. An improved adaptive genetic algorithm based on human reproduction mode for solving the knapsack problem. Inform. Technol. J., 9: 974-978.

Thomos, H.C., 2007. Introduction to Algorithm. 2nd Edn., Prentice Hall, USA., pp: 965-969.

Xiaozhen, M., X. Zhao, W. Zhao and W. Fan, 2008. Application of genetic algorithm for rectangular object layout optimisation. Proceedings of the 12th International Conference on Computer Supported Cooperative Work in Design, April 16-18, 2008, Xi'an, pp: 223-228.

Ye, Z., Z. Li and M. Xie, 2010. Some improvements on adaptive genetic algorithms for reliability-related applications. Reliab. Eng. Syst. Saf., 95: 120-126.