KMITL Sci. J. Vol. 5 No. 1 Feb. 2005

# DEMONSTRATION OF CHROMOSOME REPRESENTATIONS OF GENETIC ALGORITHMS FOR SOLVING MATHEMATICAL MODELS

Pupong Pongcharoen*

Department of Industrial Engineering, Faculty of Engineering, Naresuan University, Phitsanulok, Thailand 65000.

## ABSTRACT

This paper was aimed to demonstrate the modification on chromosome representations for solving two non-linear continuous mathematical models with single and multiple variables using binary chromosome genetic algorithms (GA). The binary chromosome was used to encode single variable and two variables models, whilst both optimum solutions from each model were initially identified. The experimental results obtained from factorial design after applying GA to solve both models, each of which with five replications, were analysed using a general linear form of analysis of variance and main effect plots. It was found that the appropriate setting of GA parameters was case dependent due to the nature of the problems and the size of its solution space. It was also found that the random seed, which is not GA parameter but is a nuisance factor occurred during the random procedure, affected on the performance of the algorithms.

**KEYWORDS:** Genetic algorithms; Design and analysis of experiment; Optimization.

---

* Corresponding author. Tel: + 66 55 261000 Ext. 4201 Fax: + 66 55 261062 Email: pupongp@nu.ac.th

## 1. INTRODUCTION

Optimization approaches can be categorized into two classes: conventional and approximation optimization algorithms [1]. Conventional optimization algorithms are usually based on mathematical models such as linear programming, branch and bound or dynamic programming. Approximation optimization algorithms are based on constructive or stochastic search techniques where the optimal solution may not be guaranteed. Constructive methods usually involve a specific rule or heuristic to construct a solution such as dispatching rules or critical path method.

Stochastic search techniques sometimes called meta-heuristics or intelligent optimization techniques [2] are characterized by a repeat process that will be terminated when a specified condition is satisfied. These algorithms such as simulated annealing, taboo search, neural network, ant colony or genetic algorithms have different mechanisms of searching an optimal solution in a solution space. Simulated annealing, taboo search and most conventional methods usually perform a unidirectional search using a single candidate solution, whilst genetic algorithms perform a multidirectional search by maintaining a number of potential solutions [3]. Due to this advantage, applied research using genetic algorithms has been found favorite.

Genetic algorithms (GA) have been used for solving various complex problems and have been successfully applied in many areas of science and engineering. The well-known applications include scheduling [4], supply chain management and logistics [5, 6], facility layout [7], course timetabling [8] and mathematical model [9]. Due to various types of GA parameters, the performance of GA may depend on the setting of its parameters such as the population size, the number of generation and the probabilities of crossover and mutation. Previous research [10] suggested that according to the nature of application problems, the appropriate setting of GA parameters and operators may be case dependent.

Since decision variables used within GA applications are encoded into chromosomes (candidate solutions), which consists of series of genes. The usage of an inappropriate coding scheme has been the cause of many GA failures [11]. Chromosome representation is therefore very important. Generally, genes are represented by either numeric (binary or real) or alphanumeric characters but the classic chromosome representation is binary digit. The objectives of this work are to demonstrate the modification on chromosome representations for solving two non-linear continuous mathematical models with single and multiple variables using binary chromosome genetic algorithms and to compare the results from both applications.

## 2. MATHEMATICAL TESTING FUNCTIONS

Two non-linear continuous mathematical models (see equation 1 and 2) adopted from Montgomery [12] and Todd [13] respectively, were used to test the performance of genetic algorithms. The first model consists of single variable ($x$), which was considered over the range between 0 to 10. The later model comprises of two variables; speed ($x_1$) and pressure ($x_2$), which were limited to the range of 100 to 140 for $x_1$ and 10 to 20 for $x_2$. The plots of both functions are illustrated by Figure 1a and 1b.

$$f(x) = 4 + [(3 - x/1.6) * (0.5 - x/1.6) * (6.2 - x/1.6) * (2 - x/1.6)]/6 \quad (1)$$

$$f(x_1, x_2) = 1217.3 - 31.256x_1 + 86.017x_2 + 0.12917x_1^2 - 2.8733x_2^2 + 0.02875x_1x_2 \quad (2)$$

(a) (b)

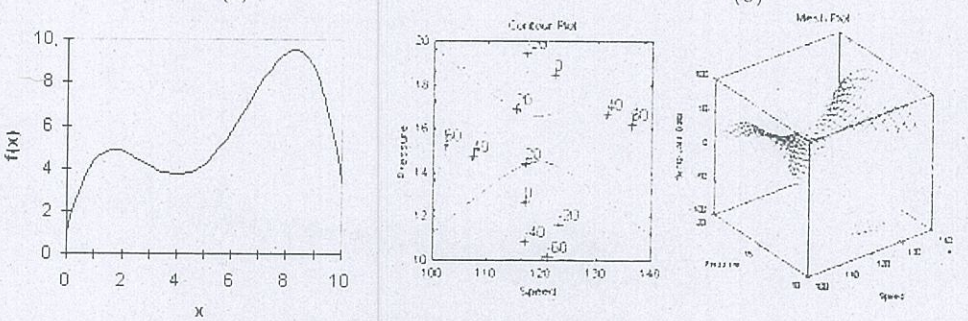


Figure 1 the plots of equation 1 and 2.

Figure 1(a) shows the plot of the equation 1 that contains both local and global optimum. The best value of $x$ at 8.29584 is initially determined and maximized the function value of 9.515504816. Figure 1(b) shows response contour and surface plots of the equation 2. The best value of $x_1$ at 119.87494 and $x_2$ at 10 were initially identified and minimized the function value of $f(x_1, x_2)$ to -66.03402.

## 3. GENETIC ALGORITHMS (GA)

Genetic algorithms, which were first presented by John Holland [14], are search and optimization tool for solving combinatorial optimization problems [15]. General procedure of GA involves chromosome representation and initialization, genetic operations (crossover and mutation), fitness evaluation and chromosome selection for the next generation [3].

Simple mechanism of genetic algorithms initially begins by encoding the problem variable to produce a series of genes, which can be represented by either numeric (binary or real), or alphanumeric characters. Genes are then randomly combined to produce a population of chromosomes, each of which represents a possible (candidate) solution. The length of the classical binary chromosome depends on the required precision. In this work, the chromosome length of ten was used to solve equation 1 and twenty (ten for each variable, $x$) for equation 2 (see Figure 2).

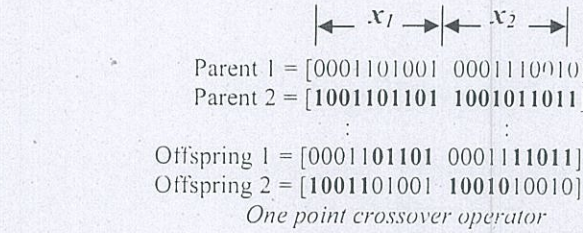


Figure 2 Crossover and mutation operations.

Genetic operations (crossover and mutation) on chromosomes are next performed by randomly selected chromosomes as parents from the population for producing offspring. There are several types of crossover and mutation operators reviewed in literature [16, 17]. In this work, one point crossover and randomly mutation operations were used (see Figure 2).

The number of parent chromosomes selected for genetic operations depend on the specification of probabilities of crossover and mutation. Todd [13] recommended that the probability of crossover should be in a range of 0.6 to 0.9 whilst the probability of mutation should range from 0.01 to 0.1. The chromosomes' fitness values are then measured using fitness (objective) function. The most famous chromosome selection process based on the roulette wheel principle is then implemented in order to select the chromosomes for the next generation. The GA process is repeated until a termination condition is satisfied.

## 4. EXPERIMENTAL DESIGN AND ANALYSIS

In this paper, experimental work was separately considered as the application of GA to solve the equation 1 (experiment A) and the equation 2 (experiment B). The design of both experiments was similar in order to draw a comparison and conclusion of the GA performance from the analysis of both experiments. Three genetic parameters considered as factors in both experiments were listed in Table 1.

| Factors considered | Level | |
|---|---|---|
| | Low (-) | High (+) |
| Population / Generation (P/G) | 40/20 (200/100) | 20/40 (100/200) |
| Probability of Crossover (%C) | 0.6 | 0.9 |
| Probability of Mutation (%M) | 0.01 | 0.18 |

Table 1 Experimental factors and its levels

The first factor was a combination of the population size and the number of generations (P/G), which influences the creation of chromosomes (candidate solutions) and execution time during the GA process. Different combinations were considered that produced a total of 800 (20x40) chromosomes for solving the equation 1 and of 20,000 (200x100) chromosomes for solving equation 2 due to larger solution space. The values considered for this factor were based on test runs that indicate convergence within the range given. The last two factors were the probabilities of crossover (%C) and mutation (%M) where the range considered for these factors were based on the recommendation by Todd [13].

The computational experiment A and B were carried out by adopting three factors and two levels full fractional factorial $2^3$ design [12] with ten replications, each of which was implemented with different random seeds. The GA program was written and run on personal computer with CPU speed of 1 GHz. Execution time for each runs on experiment A took less than ten seconds. The experimental results obtained from 80 ($2^3$*10) runs were analyzed using a general linear model form of analysis of variance provided by the statistical software package called Minitab. The analysis of variance (ANOVA) on the best function value obtained from experiment A and B are shown in Table 2 and 3, respectively. In general, the ANOVA table contains a source of variation, sum of squares (SS), degree of freedom (DF), mean squares (MS), F and $p$ values [12].

| Source | DF | SS | MS | F | $p$ |
|---|---|---|---|---|---|
| P/G | 1 | 0.0093 | 0.0093 | 3.56 | 0.063 |
| %C | 1 | 0.0034 | 0.0034 | 1.31 | 0.256 |
| %M | 1 | 0.0120 | 0.0120 | 4.60 | 0.036 |
| P/G*%C | 1 | 0.0013 | 0.0013 | 0.49 | 0.485 |
| P/G*%M | 1 | 0.0076 | 0.0076 | 2.90 | 0.093 |
| %C*%M | 1 | 0.0053 | 0.0053 | 2.01 | 0.161 |
| P/G*%C*%M | 1 | 0.0020 | 0.0020 | 0.78 | 0.382 |
| Seed | 4 | 0.0593 | 0.0148 | 5.67 | 0.001 |
| Error | 68 | 0.1778 | 0.0026 | | |
| Total | 79 | 0.2780 | | | |

Table 2 ANOVA on the best results obtained from experiment A.

| Source | DF | SS | MS | F | $p$ |
|---|---|---|---|---|---|
| P/G | 1 | 187.71 | 187.71 | 16.91 | 0.000 |
| %C | 1 | 145.73 | 145.73 | 13.12 | 0.001 |
| %M | 1 | 401.73 | 401.73 | 36.18 | 0.000 |
| P/G*%C | 1 | 38.43 | 38.43 | 3.46 | 0.067 |
| P/G*%M | 1 | 71.79 | 71.79 | 6.47 | 0.013 |
| %C*%M | 1 | 5.50 | 5.50 | 0.50 | 0.484 |
| P/G*%C*%M | 1 | 0.30 | 0.30 | 0.03 | 0.869 |
| Seed | 4 | 84.27 | 21.07 | 1.90 | 0.121 |
| Error | 68 | 755.04 | 11.10 | | |
| Total | 79 | 1690.49 | | | |

Table 3 ANOVA on the best results obtained from experiment B.

Table 2 summarized the estimation of GA parameters considered as main factors and interactions between factors corresponding to the function values obtained from the equation 1. It can be seen that only the probabilities mutation (%M) were statistically significant with a

95% confident interval for the $p$ value less than or equal to 0.05. The significant main effect plot (see Figure 3a) suggested that low level of %M should be selected. It was also found that random seed, which is not a GA parameter but is a nuisance factor occurred during the random procedure, was also statistically significant. This factor may be uncontrollable due to the random generator of each programming language. A combination of this nuisance and the nature of stochastic search within GA may create an interaction effect on the performance of GA.

From Table 3, all main factors, which were the combination of the population size and the number of generations (P/G) and the probability of crossover (%C) and mutation (%M), were statistically significant. The main effect plot (see Figure 3b) suggested that high level of P/G and %C should be selected but vice versa for %M. Moreover, only interaction between P/G and %M was found significant. Random seed was however not statistically significant in this case.

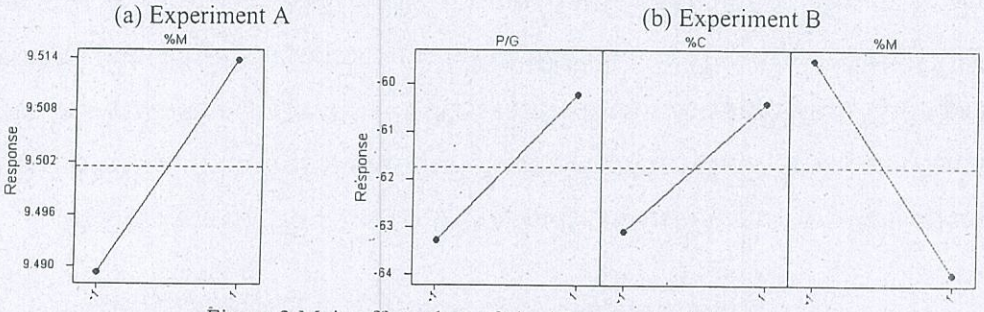


Figure 3 Main effect plots of significant main factors.

From analysis of variance and main effect plots of significant main factors of experiment A and B, it can be seen that an appropriate setting of the probability of mutation suggested by experiment A should be at low level but vice versa on experiment B. Moreover, random seed was found statistically significant on experiment A but not on experiment B. This work suggested that the appropriate setting of GA parameters was case dependent due to the different problem size and solution space.

## 5. CONCLUSIONS

Genetic Algorithms (GA) was applied to solve two non-linear continuous mathematical models with single and multiple variables. The binary chromosome was used to encode single variable and two variables models, whilst both optimum solutions from each model were initially identified. The objectives of this work were to demonstrate the modification on chromosome representations for various numbers of variables within mathematical models and comparatively discuss the results obtained from both experiments. The computational experiments were separately carried out by adopting a full factorial design with ten replications for both mathematical models.

Both experimental results obtained were analyzed using a general linear form of analysis of variance and main effect plot. It was found that only one main factor, probability of mutation, was statistically significant for both cases. Other main factors, the combination of population size and number of generations and probability of crossover, were statistically significant at least in one case. This means that the appropriate setting of GA parameters was case dependent due to the nature of the problems and the size of its solution space. It was also found that the random seed, which is not a GA parameter but is a nuisance factor occurred during the random procedure, was also statistically significant. This factor may be uncontrollable due to the random generator provided by programming language. A combination of this nuisance and the nature of stochastic search within GA may create an interaction effect on the performance of GA.

## ACKNOWLEDGEMENTS

The author would like to acknowledge the Faculty of Engineering, Naresuan University for financial support. I would also like to thank Weena Promtet, Apirak Khadwilard and Anothai Klakankhai for their help and programming support.

## REFERENCES

[1] Nagar, A., Haddock, J. and Heragu, S., **1995**. Multiple and bicriteria scheduling: a literature survey, *European Journal of Operational Research*, *81*, 88-104.

[2] Pham, D.T. and Karaboga, D., **1998**. *Intelligent Optimization Techniques Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*, Springer, London.

[3] Gen, M. and Cheng, R., **1997**. *Genetic algorithms and engineering design*, John Wiley and Sons, New York, USA.

[4] Pongcharoen, P., Hicks, C. and Braiden, P.M., **2004**. The development of genetic algorithm for the finite capacity scheduling of complex products with multiple levels of product structure, *European Journal of Production Economics*, *152*, 215-225

[5] Vergara, F.E., Khouja, M. and Michalewicz, M., **2002**. An evolutionary algorithm for optimizing material flow in supply chains, *Computers and Industrial Engineering*, *43*, 407-421.

[6] Zhang G.Q., Xue, J. and Lai, K.K., **2002**. A class of genetic algorithms for multiple level warehouse layout problems, *International Journal of Production Research*, *40*, 731-744

[7] Hicks, C., **2004**. A genetic algorithm tool for designing manufacturing facilities in the capital goods industry, *International Journal of Production Economics*, *90*, 199-211

[8] Wang, Y.Z., **2003**. Using genetic algorithm methods to solve course scheduling problems, *Expert Systems with Applications*, *25*, 39-50.

[9] De Jong, K.A., **1975**. *Analysis of Behavior of a Class of Genetic Adaptive Systems*, Doctoral dissertation, University of Michigan, Michigan.

[10] Pongcharoen, P. and Promtet, W., **2004**. Exploring and determining genetic algorithms parameters through experimental design and analysis, *Proceedings 33rd International Conference on Computers and Industrial engineering*, Jeju, Korea.

[11] Aytug H, Khouja M and Vergara F.E., **2003**. Use of genetic algorithms to solve production and operations management problems: a review, *International Journal of Production Research*, *41*(17), 3955-4009

[12] Montgomery, D.C., **1997**. *Design and Analysis of Experiments*, John Wiley & Sons, New York, USA.

[13] Todd, D., **1997**. *Multiple Criteria Genetic Algorithms in Engineering Design and Operation*, Ph.D. Thesis, Newcastle upon Tyne, UK.

[14] Holland, J.H., **1975**. *Adaptation in Natural and Artificial Systems*, University of Michigan Press.

[15] Goldberg, D.E., **1989**. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman.

[16] Nearchou, A.C., **2004**. The effect of various operators on the genetic search for large scheduling problems, *International Journal of Production Economics*, *88*, 191-203.

[17] Pongcharoen P, Stewardson D.J., Hicks C. and Braiden P.M., **2001**. Applying designed experiments to optimize the performance of genetic algorithms used for scheduling complex products in the capital goods industry, *Journal of Applied Statistics*, *28*(3&4), 441-455.