

## An Improved Genetic Algorithm for the Traveling Salesman Problem with Multi-Relations

Supat Patvichaichod

Department of Mechanical Engineering, Faculty of Engineering at Si Racha,  
Kasetsart University, 199 M.6, Tungsukhla, Si Racha, Chonburi, 20230, Thailand

---

**Abstract: Problem statement:** The aim of this research is to investigate the Traveling Salesman Problems with Multi-Relations (TSPMR) in which each of the vertices has several edges and different weights. The study concerns the weights fluctuated by time in which the problem has more complexity than general TSP problem. However, this type of problem closes to the real-world situation in engineering aspect. **Approach:** The new genetic algorithm was developed specially for the TSPMR. With the new genetic algorithm, the new coding system is a mixer between the integer numbers (represents the vertices) and the binary (represents the edges). The experiments were done to investigate the suitable input variables, which are a population size, generation, crossover rate and mutation rate. **Results:** The suitable generation between 300 and 400, the suitable crossover rate between 20 and 60% and the suitable mutation rate between 30 and 80%. **Conclusion:** The new developed method in this research is found to efficiently solve the problem with less than 100 vertices, which can reduce 50% of the transportation cost. In case of 100 -1,000 vertices problem, the transportation cost is reduced approximately by 20%. If the vertices are more than 1,000, the transportation cost can be reduced less than 20%. For the large size of problem as 30,000 vertices, the transportation cost is reduced for 6.07%.

**Key words:** Genetic algorithm, traveling salesman problem, traveling salesman problem with multi-relations (TSPMR), Generalized Chromosome Genetic Algorithm (GCGA), Gene Expression Messy Genetic Algorithm or (GEMGA), vertices, fluctuated, Hierarchical Genetic Algorithms (HGA), Successive Zooming Genetic Algorithm (SZGA)

---

### INTRODUCTION

This study presents the Traveling salesman problem with multi-relations, or TSPMR. The study concerns the case in that there is no specific number of edges among the vertices and the weights are different and vary by time. This brings to the real-world situation in engineering problems. There are various problems applied and developed from the TSPMR, for examples, the water distribution network and the heat exchanger network.

The TSPMR problem is a problem that is expanded from the past method of the TSP problem. The TSPMR problem can be considered as the NP-complete problem as well. A best solution at an appropriate time may not be determined with this approach because of a large number of the vertices in the TSPMR problem. The problem then is considered as an "Approximate algorithm". Some functions of the TSPMR method may not suitable to solve the problem that has many relations among the vertices and vary by time. So the aim of this

research is to develop the new procedures of the genetic algorithm, which is called "the Hybrid encoding genetic algorithm with multi-relations, HEGA". The HEGA is developed by using a new encoding, which mixes the binary encoding with the integer encoding. The suitable input variables can be determined from the HEGA and the experiments were done to investigate the traveling of the salesman by multi-relations that have small and large number of the vertices.

The past researches of the new encoding method are referred in various authors. The "Generalized Chromosome Genetic Algorithm (or GCGA)" was developed by Yang *et al.* (2008) to improve the encoding process in which the chromosome is divided into 2 parts, head and body, to solve the TSP problem. Kangrang and Chaleeraktragoon (2007) developed the encoding schemes that transform the monthly level of rule curves of reservoirs. The new encoding method was also presented by Carter and Ragsdale (2006) for the Multiple traveling Salesperson Problem (or mTSP). In this study, the encoding is conducted by interval

sequencing with the negative integer, which means each of the salespersons. The multi-objective genetic algorithm (MOGA) was presented by Dehuri *et al.* (2006) to develop the individual representation, to solve the data mining. The “Gene Expression Messy Genetic Algorithm or (GEMGA)” was presented by Wang and Ghosn (2005). The algorithm uses 3 types of the coding, which are 0, 1 and -1 to evaluate a reliability of the structure. Whereas the works mentioned by Acosta and Todorovich (2003) mixes the GA with the fuzzy logic by developing the new encoding called “Hierarchical Genetic Algorithms (or HGA)”. The chromosome in this algorithm has a single set of the binary to be a controller and has a set of the integer number to be a working variable. The purpose is to solve a problem of machine controlling in the industries.

The “Successive Zooming Genetic Algorithm (or SZGA)” was developed by Kwon *et al.* (2003) by using the zooming factor to improve the encoding process. The purpose is to determine a suitable value continuously. Angelov and Buswell (2003), the new encoding method was presented by using the fuzzy rule. In this study, the chromosome is divided into many parts. Each part has different value for coding. This method is suitable for the complex problem in which the old method of the encoding is not suitable. And by Abboud *et al.* (1998), the combination between the GA encoding and the Simulated Annealing (or SA) was used to solve the manpower allocation problem.

**MATERIALS AND METHODS**

**TSPMR:** Generally, the TSPMR method composes of the vertices and the edges. The traveling begins at the initial vertex and travels to each of the vertices once. Finally, the traveling returns to the initial vertex again. The purpose is to minimize the traveling cost, which is the same as general TSP problem. The method used in this research is to improve the past TSPMR problem with no specific number of the edges among the vertices. In addition, the weights of the edges are able to be changed by time.

**The procedures of the HEGA:** The procedures of the HEGA have several steps as shown in Fig. 1.

The initial population is developed by sampling the matrix of the population. The integer number with the underline means the vertices. The binary without the underline means the edges. The number of rows equals to the size of the population. The number of columns equals to the number of vertices including the number of the edges as the following example:

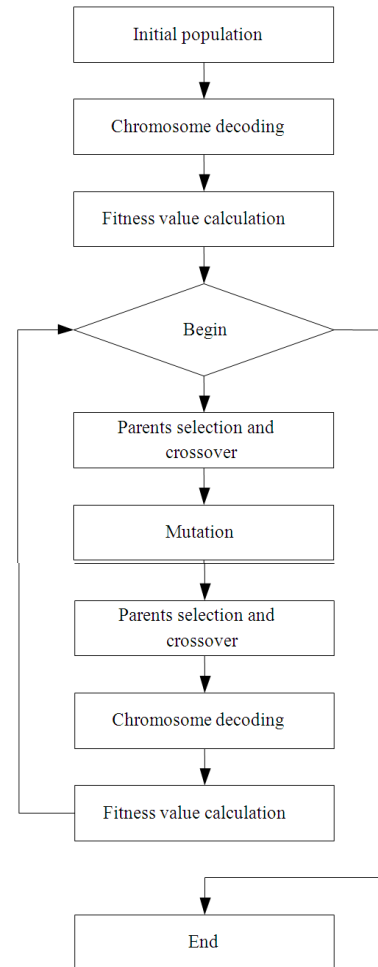


Fig. 1: Procedures of the HEGA

$$\begin{bmatrix}
 1 & 0 & 0 & \underline{2} & 1 & 1 & 1 & \underline{1} & 0 & 1 & 0 & \underline{3} & 0 & 1 & 1 \\
 0 & 1 & 1 & \underline{3} & 0 & 0 & 1 & \underline{2} & 0 & 0 & 0 & \underline{1} & 1 & 0 & 1 \\
 0 & 0 & 0 & \underline{3} & 1 & 1 & 0 & \underline{1} & 1 & 0 & 1 & \underline{2} & 0 & 1 & 1 \\
 1 & 1 & 0 & \underline{1} & 0 & 0 & 0 & \underline{3} & 1 & 1 & 1 & \underline{2} & 0 & 1 & 1
 \end{bmatrix}$$

Each row in the matrix represents each of chromosomes. Each chromosome represents the feasible solution of the problem. For examples, the first chromosome means the edge with the code of “1 0 0”, which links between the vertex at 0 (the first and final vertex) and the second vertex. The edge with the code of “1 1 1” links between the second vertex and the third vertex. The edge with the code of “0 1 0” links between the third vertex and the fourth vertex. And the edge with the code of “01 1 ” links between the fourth vertex and the vertex at 0. The matrix of the population does not

have the vertex at 0. After that, the chromosome decoding process will decode each of the edges from the binary to be the decimal number as the following example:

$$\begin{bmatrix} 4 & \underline{2} & 7 & \underline{1} & 2 & \underline{3} & 3 \\ 3 & \underline{3} & 1 & \underline{2} & 0 & \underline{1} & 5 \\ 0 & \underline{3} & 6 & \underline{1} & 5 & \underline{2} & 3 \\ 6 & \underline{1} & 0 & \underline{3} & 7 & \underline{2} & 3 \end{bmatrix}$$

The chromosome decoding is repeated by comparing with the first and final vertex. For example, among the vertex has 3 different edges and has different weights, the decoding will be as the followings.

For the integer number of 0, 1 and 2, the decoding is the first edge.

For the integer number of 3, 4 and 5, the decoding is the second edge.

And for the integer number of 6 and 7, the decoding is the third edge.

Then the matrix of the network will be as the following:

$$\begin{bmatrix} 2 & \underline{2} & 3 & \underline{1} & 1 & \underline{3} & 2 \\ 2 & \underline{3} & 1 & \underline{2} & 1 & \underline{1} & 2 \\ 1 & \underline{3} & 3 & \underline{1} & 2 & \underline{2} & 2 \\ 3 & \underline{1} & 1 & \underline{3} & 3 & \underline{2} & 2 \end{bmatrix}$$

The objective value is further calculated by using the objective function relating to each of the problem. In TSPMR problem, the low objective value is required. The following shows the example of the objective value matrix:

$$\begin{bmatrix} 46.7 \\ 34.5 \\ 32.3 \\ 27.9 \end{bmatrix}$$

From the matrix, the 4th chromosome has the lowest value of the objective value, which is 27.0. This is considered as the best chromosome. However, the rectification of the value is required to determine the fitness value. The best chromosome has the lowest objective value and is considered to be the most fitness value. The best chromosome has the most opportunity to survive for the next generation. The matrix of the fitness value has the number of rows equals to the

number of the population and has a single column as the following example:

$$\begin{bmatrix} 24.1 \\ 36.3 \\ 38.5 \\ 42.9 \end{bmatrix}$$

The next step is to select the parents by using the roulette wheel selection method and leads to the crossover. In the HEGA, the One-point crossover is used and a single cross section is selected as the following example:

$$\begin{array}{cccccccc|cccc} 0 & 1 & 1 & \underline{3} & 0 & 0 & 1 & \underline{2} & 0 & 0 & 0 & \underline{1} & 1 & 0 & 1 \\ 1 & 1 & 0 & \underline{1} & 0 & 0 & 0 & \underline{3} & 1 & 1 & 1 & \underline{2} & 0 & 1 & 1 \end{array}$$

The genes of the binary are exchanged behind the cross section. The genes of the integer number will not be exchanged as the following example:

$$\begin{array}{cccccccc|cccc} 0 & 1 & 1 & \underline{3} & 0 & 0 & 1 & \underline{2} & 0 & 1 & 1 & \underline{1} & 0 & 1 & 1 \\ 1 & 1 & 0 & \underline{1} & 0 & 0 & 0 & \underline{3} & 1 & 0 & 0 & \underline{2} & 1 & 0 & 1 \end{array}$$

After the selection and crossover processes, the next step of evolution of the mutation is begun in which each of chromosomes is operated. In the HEGA, the Exchange mutation is used by sampling the 2 genes from the vertices and the values are exchanged as the following example:

$$\begin{array}{cccccccccccc} 1 & 0 & 0 & \underline{2} & \underline{1} & \underline{1} & \underline{1} & \underline{1} & 0 & 1 & 0 & \underline{3} & 0 & 1 & 1 \\ 1 & 0 & 0 & \underline{1} & \underline{1} & \underline{1} & \underline{1} & \underline{2} & 0 & 1 & 0 & \underline{3} & 0 & 1 & 1 \end{array}$$

The next evolution to a new generation is then begun and the evolution process continues to the defined generation, which is the end of the HEGA.

**Research method:** The procedures in this research start with setting the 14 TSPMR problems, which starts from the smallest number of the vertices to the largest number of the vertices. The set of the number of vertices is composed of 20, 40, 60, 80, 100, 200, 400, 600, 800, 1,000, 1,500, 2000, 500 and 3, 000 respectively. The experiments were conducted to determine a suitable value of the input variables to find the preferable solution under the acceptable time interval. The input variables under investigation are

composed of population size (N), the probability of the crossover (Pc), the probability of the mutation (Pm) and the Generation (G). For all of the 14 problems, the Pc, Pm and G are set to be constant at the beginning to determine a suitable value of N. After that the N, Pc and Pm are set to be constant to determine the appropriate G by the problem size. Finally, the N and G value are set to be constant to determine the appropriate value of Pc and Pm.

The experiments measure the efficiency of the HEGA by 3 methods. The first method measures the percentage of the traveling cost saving as the following equation:

$$\text{Cost saving} = [(ACF-MCL)/ACF] \times 100\% \quad (1)$$

Where:

ACF = Average cost of the first generation

MCL = Minimum cost of the last generation

The second method measures the percentage of the convergence rate as the following equation:

$$\text{Convergence rate} = [(MCF-MCL)/MCF] \times 100\% \quad (2)$$

where, MCF, minimum cost of the first generation

The third method measures the run time of the HEGA in a second unit.

### RESULTS AND DISCUSSION

The experimentation to determine a suitable value of N by the problem size was conducted by varying the N values that relate to the problem size whereas the input variables are set to be constant. The Pc value is 70%, the Pm value is 5% and the G value is 500 generations. Each experiment is run for 5 times. The result showed that the N value should be a half of the number of the vertices in case of the TSPMR problem has less than 100 vertices. For the population size of the problem that has more than 100 vertices but no more than 1,000 vertices, the N value should be one-tenth of the number of vertices. And for the population size of the problem that has more than 1,000 vertices, the N value should be one-twentieth of the number of vertices. For the TSPMR problem that has less than 60 vertices, the generations should be 200. For the TSPMR problem that has more than 60 vertices but no more than 1,000 vertices, the generations should be 300. And for the TSPMR problem that has more than

1,000 vertices, the generations should be 400. The Probability of Crossover (Pc) of the TSPMR problem that has less than 100 vertices should have a percentage during 20-60%. The probability of the crossover of the TSPMR problem that has more than 100 vertices should have the percentage during 30-50%. The suitable Pc of this developed method is different from the suitable Pc of general genetic algorithm, which has the percentage during 95-60%.

The probability of the mutation of the TSPMR problem that has less than 600 vertices should have the percentage during 30-70%. The probability of the crossover of the TSPMR problem that has more than 600 vertices should have the percentage during 60-80%. Fig. 2 and Fig. 3 are the examples of the relationship between Pm and Pc for 1,000 vertices problem. The convergence rate depends on the traveling cost saving. The run time does not depend on the probability of crossover and the probability of mutation. However, the run time depends on the problem size, population size and number of generation.

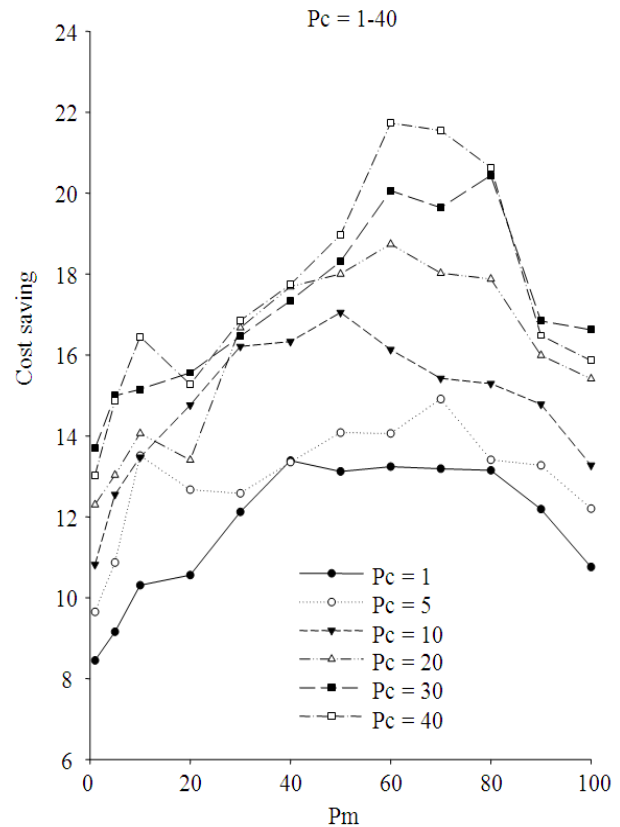


Fig. 2: The relationship between Pm during 1-100% and Pc during 1-40% for 1,000 vertices problem

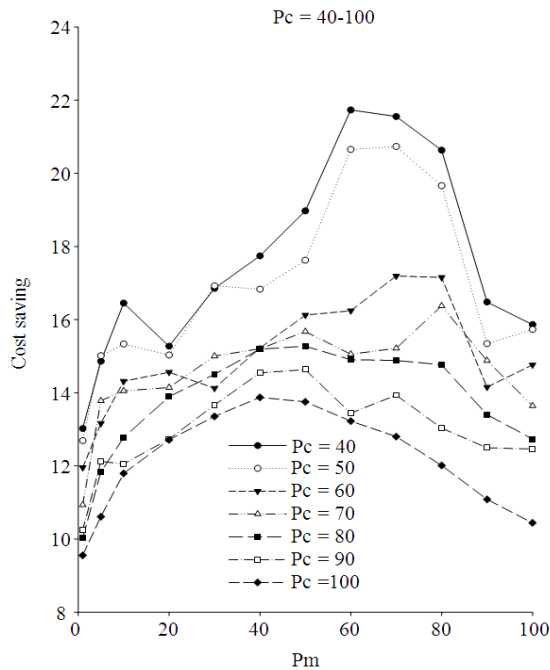


Fig. 3: The relationship between Pm during 1-100% and Pc during 40-100 % for 1,000 vertices problem

**CONCLUSION**

The Hybrid Encoding Genetic Algorithm with multi-relations (or HEGA) that was developed in this research has ability to solve for an appropriate solution of the TSPMR problem efficiently. The reduction in traveling cost reverses with the problem size. For examples, the problem with less than 100 vertices, the traveling cost can be reduced by 50%. The problem of the salesperson with multi-relations and has 100-1,000 vertices, the traveling cost can be reduced by 20%. And the problem of the salesperson with multi-relations with more than 1,000 vertices, the traveling cost can be reduced less than 20%. For large scale of problem that has 30,000 vertices, the traveling cost can be reduced by 6.07%. Although the percentage of traveling cost reduction of the large problem is less than the small size problem, but the reduction of money value of the large problem is more than small size problem. This is because the large problem has higher initial cost.

The results are also shown that the preferable convergence rate depends on the saving traveling cost. The run time does not depend on the probability of crossover and the probability of mutation but depends on the problem size, population size and number of generations.

The suitable probability of mutation of HEGA is different from the suitable probability of mutation of

general genetic algorithm between 60 and 95%. And the suitable probability of crossover of HEGA is different from the suitable probability of crossover of general genetic algorithm 0 and 5% whereas the suitable input variables are different by the problem size. The future research of the genetic algorithm should concern the selection of the mutation and the improvement of the mutation that leads to better solution. The HEGA method should be more applied with the engineering problems.

**REFERENCES**

Abboud, N., M. Inuiguchi, M. Sakawa and Y. Uemura, 1998. Manpower allocation using genetic annealing. *Eur. J. Opera. Res.*, 111: 405-420. DOI: 10.1016/S0377-2217(97)00326-3

Acosta, G. and E. Todorovich, 2003. Genetic algorithms and fuzzy control: a practical synergism for industrial applications. *Comput. Ind.*, 52: 183-195. DOI: 10.1016/S0166-3615(03)00102-7

Angelov, P.P. and R.A. Buswell, 2003. Automatic generation of fuzzy rule-based models from data by genetic algorithms. *Inform. Sci.*, 150: 17-31. DOI: 10.1016/S0020-0255(02)00367-5

Carter, A.E. and C.T. Ragsdale, 2006. A new approach to solving the multiple traveling salesperson problem using genetic algorithms. *Eur. J. Opera. Res.*, 175: 246-257. DOI: 10.1016/j.ejor.2005.04.027

Dehuri, S., A.K. Jagadev, A. Ghosh and R. Mall, 2006. Multi-objective genetic algorithm for association rule mining using a homogeneous dedicated cluster of workstations. *Am. J. Applied Sci.*, 3: 2086-2095. DOI: 10.3844/2006.2086.2095

Kangrang, A. and C. Chaleeraktragoon, 2007. Genetic algorithms connected simulation with smoothing function for searching rule curves. *Am. J. Applied Sci.* 4: 73-79. DOI: 10.3844/2007.73.79

Kwon, Y.D., S.B. Kwon, S.B. Jin and J.Y. Kim, 2003. Convergence enhanced genetic algorithm with successive zooming method for solving continuous optimization problems. *Comp. Struct.*, 81: 1715-1725. DOI: 10.1016/S0045-7949(03)00183-4

Wang, J. and M. Ghosn, 2005. Linkage-shredding genetic algorithm for reliability assessment of structural systems. *Struc. Safety*, 27: 49-72. DOI: 10.1016/j.strusafe.2004.06.001

Yang, J., C. Wua, H.P. Lee and Y. Liang, 2008. Solving traveling salesman problems using generalized chromosome genetic algorithm. *Progress Natural Sci.*, 18: 887-892. DOI: 10.1016/j.pnsc.2008.01.030