

Application of Hybrid Encoding Genetic Algorithm on Pickup and Delivery Traveling Salesman Problem with Traffic Conditions

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: This study deals with the pickup and delivery traveling salesman problem with traffic conditions (PDTSPTW), an extension of the pickup and delivery traveling salesman problem (PDTSP) where each customer to be served is associated with two quantities of product to be collected and delivered. Almost PDTSP problems uses distance between each point of customers as Euclidean Distance and are not concerned with other parameters to find minimal cost. **Approach:** The PDTSPTC concerns more parameters, such as street network and vehicle speed, which results it closer to the real world condition. The study also proposes the developed genetic algorithm called "Hybrid Encoding Genetic Algorithm (HEGA)". The concept is to combine binary encoding and integer encoding together, causing the in complexity of the algorithm structure and the ease of implementation. **Results:** the HEGA can save the travel cost about 26-57%. It is obviously seen that HEGA in the PDTSPTC test problems result the fast convergence that is about 12-43% and in all, the computational time is under 6 seconds. **Conclusion:** The HEGA performs quite well when testing a test problem. Computation times are small in all case. The PDTSPTC is closer to real-world conditions than PDTSP. This problem can be applied in logistics immediately if the distance of streets, traffic conditions (average vehicle speed in each street) and vehicle conditions (fuel consumption rate and vehicle capacity) are known.

Key words: Hybrid encoding, genetic algorithm, traveling salesman problem, Traveling Salesman Problem with Multi-Relations (TSPMR), Pickup and Delivery Traveling Salesman Problem (PDTSP), multiple Traveling Salesperson Problem (m-TSP), traffic conditions, vehicle conditions, Vehicle Routing Problem (VRP)

INTRODUCTION

The purpose of this study is to describe the new pickup and delivery traveling salesman problem (PDTSP) called Pickup and Delivery Traveling Salesman Problem with Traffic Condition (PDTSPTC). In PDTSP, there are two types of customers, each one with a given demand and a vehicle with a given capacity, originally stationed in the depot. Also travel distances are given. The product collected from pickup customers is different from the product supplied to delivery customers. Moreover, the total amount of product collected from pickup customers must be delivered only to the depot and the product collected from the depot must be delivered to the delivery customers. For example, this is the case when empty bottles must be collected from customers to a warehouse and full bottles must be delivered from the warehouse to the customers.

The novelty of the PDTSPTC respects to the PDTSP is the real world parameters. Almost proposed

PDTSP problems concern only distance between points of customers (Euclidean Distance). However to reduce the cost of transportation in the real world, more parameters such as street network, traffic conditions and vehicle conditions have to be studied. In PDTSPTC, all customers are connected by street network. The distance between points of customer is not always a Euclidean distance. Moreover, the traffic condition in each particular street in each particular time of traveling is set by varying the average vehicle speed. Finally, the vehicle condition is represented by fuel consumption load that is depended on the average vehicle speed and vehicle capacity.

PDTSPTC are defined as follows. Let $G = (V_0, R)$ be an undirected graph where $V_0 = \{0\} \cup V$ is the customer set (with $V = \{1, \dots, n\}$). Vertex 0 represents the depot, whereas the other vertices $i \in V$ represent customers, with an associated delivery quantity, $d_i \geq 0$ and pickup quantity, $p_i \geq 0$, (with $d_0 = p_0 = 0$). For each route $(i, j) \in R$ and $i, j \in V_0$ a non-negative traveling cost, C_{ij} , is also given. A vehicle with capacity Q is stationed

at the depot. The PDTSPTC consists of determining a tour starting and ending at the depot, serving each customer exactly once and having minimum traveling cost. The total load of the vehicle along the tour may never exceed the vehicle capacity, Q . Moreover, if a customer requires both delivery and pickup, the two operations must not be serviced separately and the delivery operation is assumed to be performed first.

The PDTSPTC is NP-Hard in the strong sense since it generalizes the well-known NP-Hard Traveling Salesman Problem with Traffic Conditions (TSPTC), arising when for all Relatively few algorithms have been proposed for the PDTSP and related topics. Thongwan *et al.* (2011), Mahi and Izabatene (2011), Mosavi (2011), Asfaw and Saiedi, (2011), Norozi *et al.* (2010), Srichandum and Rujirayanyong (2010), Yedjour *et al.* (2010), Kangrang *et al.* (2009), Renaud *et al.* (2002), Gendreau *et al.* (1996; 1999) and Kalantari *et al.* (1985). This study proposes the developed genetic algorithm (GA) for PDTSPTC, called “Hybrid Encoding Genetic Algorithm (HEGA)”. It is a Meta-Heuristics method that can find the good solution quality using a reasonable amount of time. The past researches of the new encoding method are referred in various authors. Patvichaichod (2011), Nazif and Lee (2010), Sarabian and Lee (2010), Al Rahedi and Atoum (2009), Yang *et al.* (2008), Kangrang and Chaleeraktrakoon (2007) and Dehuri *et al.* (2006).

MATERIALS AND METHODS

The test problems: As there are no test problems of PDTSPTC in the literature, thus the test problem of PDTSPTC was developed, it has single depot (D). The problem was set to 8 customers (C1-C8) and 27 streets (S1-S27) as shown in Fig. 1. The vehicle starts from the depot at 8.00 a.m., visits all customers just once then pickups and/or deliveries of product and finally goes back to the depot.

Because each particular street has particular traffic condition, depending on the traveling time, the average speed of the vehicle (in term of kilometers per hour, km h^{-1}) in each particular street is varied by the time of traveling. In the test problem, the traffic conditions of all streets are set to change at 12.00 a.m. causing the change of average speed of the vehicle in each street.

Fuel consumption rate of the vehicle is set by varying on the average speed and the vehicle capacity. The minimum fuel consumption rate occurs at the speed of 50 and 60 km h^{-1} in all vehicle capacity, as shown in Fig. 2.

Because all streets are connected as a network, the number of possible routes between 2 customers is large. The possible routes from customer 1 to customer 3 can

be defined as S13 - S15, S13 - S14 - S17 - S16, S13 - S14 - S7 - S19 - S18 - S16, S11 - S10 - S12 - S15, S11 - S10 - S12 - S15, S11 - S10 - S12 - S14 - S17 - S16, S11 - S10 - S5 - S3 - S6 - S17 - S16, S11 - S10 - S5 - S1 - S2 - S7 - S17 - S16 and so on. The search space is very large. Thus the number of sets of streets between customer 1 and customer 3 is reduced to 2 (S13 - S15 and S13 - S14 - S17 - S16) by defining these 2 routes as feasible routes and others as infeasible routes. Finally, the travel cost in term of Thai Baht is evaluated. When the possible routes are defined, the streets in routes can be obtained and the travel can be calculated by using the distance, the average speed and the fuel consumption rate. The used calculation methods are simple such as speed equal distance divided by time, travel cost equal distance multiple by fuel consumption rate. Because PDTSPTC is classified as NP-Hard problem that cannot be solved by deterministic algorithms within an acceptable period of time, since it has numerous local minima.

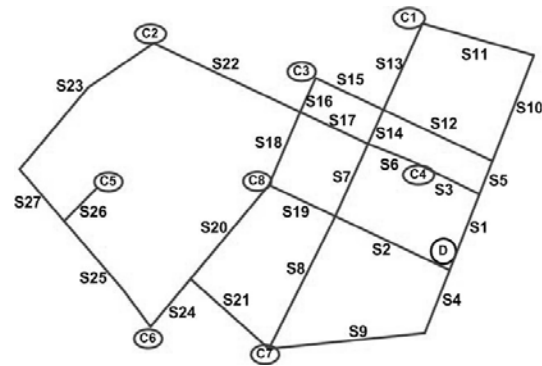


Fig. 1: Test problem

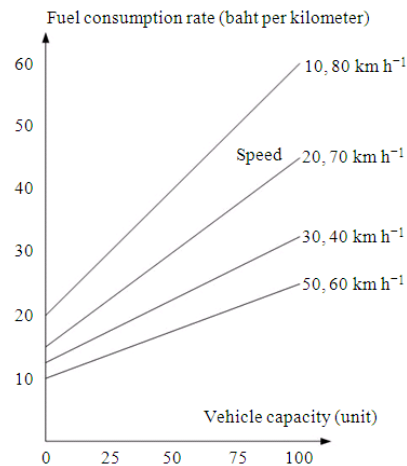


Fig. 2: Fuel consumption rate depends on the average speed and vehicle capacity

Genetic Algorithm (GA) are well suited to solve complicated and multi-variable optimization problems (Chen and Tseng, 1996). GA are search strategies ideally suited to parallel computing and most effectively applied to problems in which small changes result in very nonlinear behavior in the solution space (Kamhawi *et al.*, 1996).

Hybrid encoding genetic algorithm: GA are Meta-Heuristic inspired by the efficiency of natural selection in biological evolution. Unlike traditional heuristics (and some Meta-Heuristic like tabu search) that generate a single solution and work hard to improve it, GA maintain a large number of solutions and perform comparatively little work on each one. The collection of solutions currently under consideration is called the population. Each member of the population (called a chromosome) is an encoded version of a solution. The encoding strategy is different for different optimization problems and a given problem may have more than one workable encoding strategy. The goal of any encoding is to translate a solution into a string of genes that make up the chromosome, just as in biological genetics. Each iteration of a GA consists of several operators that construct a new generation of solutions from the old one in a manner designed to preserve the genetic material of the better solutions (survival of the fittest). Many GA operators have been proposed; the three most common are reproduction, crossover and mutation. Reproduction consists of simply copying the best solutions from the previous generation into the next, with the intention of preserving very high-quality solutions in the population as-is. Crossover takes two “parents,” randomly chosen and produces one or more “offspring” that contain some combination of genes from the parents. Crossover can be performed in a deterministic manner (e.g., “one-point” crossover), with genes appearing before a certain cutoff coming from parent 1 and genes after the cutoff coming from parent 2, or in a random manner, with each gene taken from a given parent with a certain probability. The mutation operator changes a few genes randomly, borrowing from the evolutionary concept that random genetic mutations may produce superior offspring (or, of course, inferior offspring, but such individuals are less likely to survive from one generation to the next). Because the standard GA are not suitable to implement PDTSPCT. Thus in this study, new GA, called “Hybrid Encoding Genetic Algorithm (HEGA)” is developed for the problem of PDTSPCT.

The HEGA merges binary encoding that represent streets and integer encoding that represent customers. The creation of a new generation of chromosomes involves primarily four major steps: selection, crossover, mutation and reproduction. In the selection step, the roulette wheel selection was employed. In crossover step, only the one-point crossover for binary encoding was used. In mutation step, only the exchange mutation for integer encoding was performed and in reproduction step, the best chromosome is copied from the previous generation to the next one. The HEGA can be modified easily to handle PDTSPCT problem. The general structure of the HEGA can be described as follows:

```

Procedure HEGA algorithm for PDTSPCT
    Generation = 1;
    initialize population;
    evaluate fitness;
    while (termination condition not satisfied) do
        generation = generation + 1;
        select;
        crossover;
        mutate;
        reproduce;
        evaluate fitness;
    end
End HEGA algorithm for PDTSPCT.

```

Initial population: The initial population is randomly generated. The gray boxes represent customers (1, 2 and 3) and alphabet D in the first and last gray boxes represent Depot. The binary in the white boxes represent streets, as shown in Fig. 3.

Decoding: When the initial population is generated, the algorithm decodes all chromosomes into sets of streets between each customer. For example, chromosome 1 in Fig. 5, the first customer from depot is customer 3. Binary encoding between depot and customer 3 is “0-0-1” decoding to “1” of decimal numeral system and the set of streets S1-S5-S12-S15. The second customer is customer 1, binary encoding between customer 3 and customer 1 is “0-0-0” decoding to “0” of decimal numeral system and the set of streets S15-S13.

Chromosome 1	D	0	0	1	3	0	0	0	1	0	0	0	2	0	1	1	D
Chromosome 2	D	1	0	0	2	1	0	1	3	1	0	1	2	1	1	1	D
Chromosome 3	D	1	1	1	3	0	1	1	2	0	0	0	1	1	0	1	D
Chromosome N	D	0	0	1	1	1	0	1	2	1	1	1	3	0	1	0	D

Fig. 3: Population

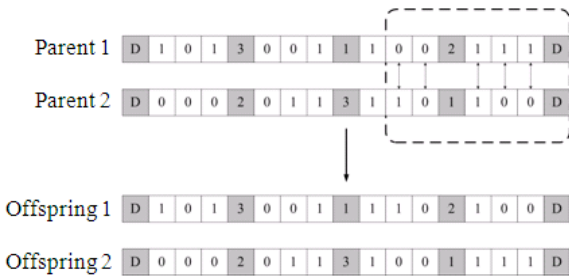


Fig. 4: One-point crossover

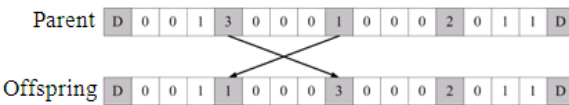


Fig. 5: Exchange mutation

The third customer is customer 2, binary encoding between customer 1 and customer 2 is “0-0-0” decoding to “0” of decimal numeral system and the set of streets S13-S14-S17-S22. When the vehicle visits all customer and return to the depot, binary encoding between customer 2 and depot is “0-1-1” decoding to “3” of decimal numeral system and the set of streets S22-S18-S19-S2.

Selection operator: The selection operator is an important part in GA. This part is independent of other parts. It has no direct relation to the problem and the fitness function, crossover and mutation operators. There are many different selection operators presented by researchers, such as the elitist model, tournament selection, deterministic sampling, remainder stochastic sampling without replacement, stochastic tournament and roulette wheel selection. These selection operators are employed for problems of numerical optimization. The main objective is to reduce the sampling error and improve calculating precision. In the HEGA, tournament selection is employed. If some chromosomes have the capacity more than maximum capacity of the vehicle, the fitness value of those chromosomes will be increased and not survive to next generation.

Crossover operator: The role of this operator is to join together parts of several individuals in order to produce new two individuals for the next generation. The individuals are randomly selected according to a predefined probability (crossover rate). In HEGA we use one-point crossover to generate offspring. First, two parents are chosen at random. Two offsprings are generated from the two parents; only the interchange for binary encoding is used, as shown in Fig. 4.

Mutation operator: Inconsistent mutation is adopted in process sequencing HEGA. Some individuals of the next generation which are obtained through the two above-mentioned operators are selected and then exchange the positions of two integer encoding in each individual randomly to realize the mutation operation. For example, the second and third gray box are the selected mutation positions in parent, the code in second position is “3” and “1” is in third position, exchange code “3” and “1” to finish the mutation operation. New individual (offspring) will be obtained and will replace parents in the next generation, as shown in Fig. 5.

Reproduction operator: Reproduction is an operator that follows the process of natural selection and the “survival of the fittest”. It is a simple copy of an individual from previous generation to the next generation without any modification. Reproduction has the purpose of preserving the good traits, carried by the good individuals of the population and spreading them over the population at a higher rate. Consequently, reproduction does not produce new individuals. In the HEGA, the best chromosome is copied from the previous generation to the next one according to the normalized (and not the absolute) values of the fitness function. The proportional selection is based on the roulette wheel strategy, where the chromosomes that will be copied are selected with rates proportional to their fitness. This means that the probability is higher for a chromosome with high fitness to be selected for reproduction than another with lower fitness.

RESULTS

In the experiment, the HEGA is implemented in Visual Basic program and tested on a laptop computer with a 2.0 GHz processor and 2 GB RAM, running under Windows XP. The parameters, needed to input for running HEGA program, are population size (N), probability of crossover (Pc), probability of mutation (Pm) and number of generation (G). N is fixed to 10, Pc is set to 0.4, Pm is set to 0.5 and G is fixed to 200. The test problem is breakup into 16 sub-problems that are varying from 5 to 8 customers. The delivery and pickup quantity of each customer is set to varying in each sub-problem. The positive value of product quantity is represents the pickup demand of customers and the negative value is represents the delivery demand. The maximum capacity of the vehicle is set to 100 units of product.

Table 1: The results of 16 sub-problems that are varying from 5 to 8 customers.

No. of customers	Pickup and delivery capacity of customers								Aver. cost (baht)	Min. cost (baht)	Travel cost saving (%)	Conv. rate	Time (sec)
	C1	C2	C3	C4	C5	C6	C7	C8					
5	+0,-20	+25,-0	+0,-30	+10,-10	+25,-0	-	-	-	15021	7632	49.2	35.4	4.5
	+10,-15	+20,-10	+5,-35	+30,-0	+0,-20	-	-	-	15355	9112	40.7	25.0	3.9
	+0,-35	+15,-0	+5,-15	+20,-10	+10,-0	-	-	-	13205	6617	49.9	33.8	5.1
	+15,-0	+0,-20	+15,-0	+0,-25	+0,-30	-	-	-	15559	7413	52.4	40.0	4.9
6	+0,-20	+25,-0	+0,-30	+10,-10	+25,-0	+10,-10	-	-	16230	7556	53.4	42.0	5.2
	+10,-15	+20,-10	+5,-35	+30,-0	+0,-20	+15,-10	-	-	16673	8312	50.1	38.3	3.8
	+0,-35	+15,-0	+5,-15	+20,-10	+10,-0	+0,-30	-	-	13208	8166	38.2	18.3	4.6
	+15,-0	+0,-20	+15,-0	+0,-25	+0,-30	+20,-0	-	-	18230	9087	50.2	39.5	4.5
7	+0,-20	+25,-0	+0,-30	+10,-10	+25,-0	+10,-10	+15,-0	-	17365	8043	53.7	43.2	4.9
	+10,-15	+20,-10	+5,-35	+30,-0	+0,-20	+15,-10	+10,-0	-	20203	10664	47.2	37.2	5.3
	+0,-35	+15,-0	+5,-15	+20,-10	+10,-0	+0,-30	+20,-10	-	14476	8219	43.2	27.0	5.1
	+15,-0	+0,-20	+15,-0	+0,-25	+0,-30	+20,-0	+10,-15	-	19430	14256	26.6	12.1	5.3
8	+0,-20	+25,-0	+0,-30	+10,-10	+25,-0	+10,-10	+15,-0	+0,-15	16599	7090	57.3	47.0	6.0
	+10,-15	+20,-10	+5,-35	+30,-0	+0,-20	+15,-10	+10,-0	+0,-10	18430	8766	52.4	42.4	5.4
	+0,-35	+15,-0	+5,-15	+20,-10	+10,-0	+0,-30	+20,-10	+25,-0	17340	9510	45.2	32.7	5.7
	+15,-0	+0,-20	+15,-0	+0,-25	+0,-30	+20,-0	+10,-15	+5,-10	18321	9890	46.0	34.6	5.5

In this experiment, the performance of HEGA algorithm is evaluated by comparing the minimum travel cost at last generation (MTC_L) and average travel cost at first generation (ATC_F). The saving of the travel cost using HEGA to solve PDTSPTC can be calculated as:

$$\text{Travel cost saving} = (ATC_F - MTC_L) / ATC_F$$

The other criterion is used to measure the performance of HEGA for solving PDTSPTC is convergence rate. It is to compare the MTC_L and the minimum travel cost at first generation (MTC_F) that show how many rate of HEGA is converged, by using the following formula:

$$\text{Convergence rate} = (MTC_F - MTC_L) / MTC_F$$

Table 1 report the travel cost saving, convergence rate and computational time of 16 sub-problems. Results reported indicate that the HEGA can save the travel cost about 26-57%. It is obviously seen that HEGA in the PDTSPTC test problems result the fast convergence that is about 12-43% and in all, the computational time is under 6 sec.

DISCUSSION

The difficulty of PDTSPTC are how to define the average speed of the vehicle in each street that varied by time and the number of feasible routes between each customers.

CONCLUSION

This study presents a new genetic algorithm called “Hybrid Encoding Genetic Algorithm” (HEGA) to

solve a Pickup and Delivery Traveling Salesman Problem with Traffic conditions (PDTSPTC). The HEGA algorithm incorporates binary encoding and integer encoding together. The algorithm performs quite well when testing a test problem. Computation times are small in all case. The PDTSPTC is closer to real-world conditions than PDTSP. This problem can be applied in logistics immediately if the distance of streets, traffic conditions (average vehicle speed in each street) and vehicle conditions (fuel consumption rate and vehicle capacity) are known. Considering the future research work, the PDTSPTC can be extended to consider time-windows and can also be extended to a new concept of Vehicle Routing Problem (VRP). The HEGA can be developed to use new crossover and mutation schemes for a better solution to solve PDTSPTC and the future research work.

REFERENCES

Al Rahedi, N.T. and J. Atoum, 2009. Solving the traveling salesman problem using new operators in genetic algorithms. *Am. J. Applied Sci.*, 6: 1586-1590. DOI: 10.3844/ajassp.2009.1586.1590

Asfaw, T.D. and S. Saiedi, 2011. Optimal short-term cascade reservoirs operation using genetic algorithm. *Asian J. Applied Sci.*, 4: 297-305. DOI:10.3923/ajaps.2011.297.305

Chen, C.J. and C.S. Tseng, 1996. The path and location planning of workpieces by genetic algorithms. *J. Intell. Manuf.*, 7: 69-76. DOI: 10.1007/BF00114139

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/ajassp.2006.2086.2095

- Gendreau, M., A. Hertz and G. Laporte, 1996. The traveling salesman problem with backhauls. *Comput. Operat. Res.*, 23: 501-508. DOI: 10.1016/0305-0548(95)00036-4
- Gendreau, M., G. Laporte and D. Vigo, 1999. Heuristics for the traveling salesman problem with pickup and delivery. *Comput. Operat. Res.*, 26: 699-714. DOI: 10.1016/S0305-0548(98)00085-9
- Kalantari, B., A.V. Hill and S.R. Arora, 1985. An algorithm for the traveling salesman problem with pickup and delivery customers. *European J. Operat. Res.*, 22: 377-386. DOI: 10.1016/0377-2217(85)90257-7
- Kamhawi, H.N., R.S. Leclair and C.L.P. Chen, 1996. Feature sequencing in the rapid design system using a genetic algorithm. *J. Intell. Manuf.*, 7: 55-67. DOI: 10.1007/BF00114138
- 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/ajassp.2007.73.79
- Kangrang, A., S. Compliew and W. Chaiyapoom, 2009. Heuristic algorithm with simulation model for searching optimal reservoir rule curves. *Am. J. Applied Sci.*, 6: 263-267. DOI: 10.3844/ajassp.2009.263.267
- Mahi, H. and H.F. Izabatene, 2011. Segmentation of satellite imagery using RBF neural network and genetic algorithm. *Asian J. Applied Sci.*, 4: 186-194. DOI: 10.3923/ajaps.2011.186.194
- Mosavi, M.R., 2011. Applying genetic algorithm to fast and precise selection of GPS satellites. *Asian J. Applied Sci.*, 4: 229-237. DOI: 10.3923/ajaps.2011.299.237
- Nazif, H. and L.S. Lee, 2010. Optimized crossover genetic algorithm for vehicle routing problem with time windows. *Am. J. Applied Sci.*, 7: 95-101. DOI: 10.3844/ajassp.2010.95.101
- Norozi, A., M.K.A. Ariffin and N. Ismail, 2010. Application of intelligence based genetic algorithm for job sequencing problem on parallel mixed-model assembly line. *Am. J. Eng. Applied Sci.*, 3: 15-24. DOI: 10.3844/ajeassp.2010.15.24
- Patvichaichod, S., 2011. An improved genetic algorithm for the traveling salesman problem with multi-relations. *J. Comput. Sci.*, 7: 70-74. DOI: 10.3844/jcssp.2011.70.74
- Renaud, J., F.F. Boctor and G. Laporte, 2002. Perturbation heuristics for the pickup and delivery traveling salesman problem. *Comput. Operat. Res.*, 29: 1129-1141. DOI: 10.1016/S0305-0548(00)00109-X
- Sarabian, M. and L.V. Lee, 2010. A modified partially mapped multicrossover genetic algorithm for two-dimensional bin packing problem. *J. Math. Stat.*, 6: 157-162. DOI: 10.3844/jmssp.2010.157.162
- Srichandum, S. and T. Rujirayanyong, 2010. Production scheduling for dispatching ready mixed concrete trucks using bee colony optimization. *Am. J. Eng. Applied Sci.*, 3: 7-14. DOI: 10.3844/ajeassp.2010.7.14
- Thongwan, T., A. Kangrang and S. Homwuttiwong, 2011. An estimation of rainfall using fuzzy set-genetic algorithms model. *Am. J. Eng. Applied Sci.*, 4: 77-81. DOI: 10.3844/ajeassp.2011.77.81
- Yang, J., C. Wua, H.P. Lee and Y. Liang, 2008. Solving traveling salesman problems using generalized chromosome genetic algorithm. *Prog. Nat. Sci.*, 18: 887-892. DOI: 10.1016/j.pnsc.2008.01.030
- Yedjour, D., H. Yedjour and A. Benyettou, 2010. Combining quine mc-cluskey and genetic algorithms for extracting rules from trained neural networks. *Asian J. Applied Sci.*, 4: 72-80. DOI: 10.3923/ajaps.2011.72.80