Kasetsart J. (Nat. Sci.) 42 : 387 - 395 (2008)

Pickup and Delivery Traveling Salesman Problem with Traffic Conditions

Supat Patvichaichod[1]* and Poranat Visuwan[2]

ABSTRACT

This paper deals with the pickup and delivery traveling salesman problem with traffic conditions (PDTSPTC), 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. The PDTSPTC concerns more parameters, such as street network and vehicle speed, which results it closer to the real world condition. The paper also proposes the new developed genetic algorithm called "Hybrid Encoding Genetic Algorithm (HEGA)". The concept is to combine binary encoding and integer encoding together, causing the incomplexity of the algorithm structure and the ease of implementation. The main advantage of this new algorithm is the improvement of overall system efficiency.

Key words: traveling salesman problem, pickup and delivery, genetic algorithm and artificial intelligent

INTRODUCTION

The purpose of this paper 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

1 Faculty of Engineering at Si Racha, Kasetsart University, Si Racha Campus, Chon Buri 20230, Thailand.
2 Faculty of Engineering, Kasetsart University, Bangkok 10900, Thailand.
* Corresponding author, e-mail: supat@src.ku.ac.th

Received date : 16/07/07 Accepted date : 29/10/07

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 $V=\{1,...,n\}$). 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 Q capacity 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) (Patvichaichod and Visuwan, 2007), arising when $p_i = d_i$ for all $i \in V$. Relatively few algorithms have been proposed for the PDTSP. Kalantari *et al*. (1985) and Perez and Gonzalez (2004) have described an exact branch-and-bound and branch-and-cut procedure for the PDTSP, respectively. Savelsbergh (1990), Healy and Moll (1995), Gendreau *et al*. (1996), Rosenkrantz *et al*. (1997), Gendreau *et al*. (1999), Renaud *et al*. (2000) and Renaud *et al*. (2002) have porposed several heuristic procedure for the PSTSP. Glover and Laguna (1997) have described a tabu search for the PDTSP. This paper 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.

MATERIALS AND METHODS

Part I. 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 Figure 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/hr) 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. (Figure 2).

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/hr in all vehicle capacity, as shown in Figure 3.

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 -

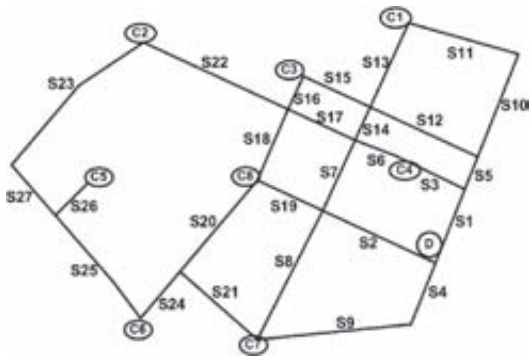

Figure 1 Test problem.

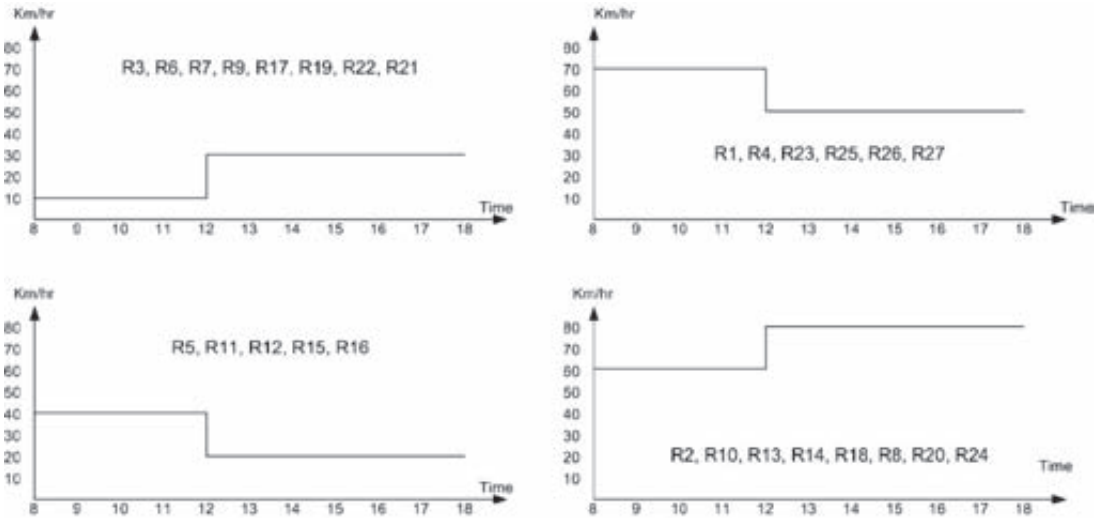

Figure 2 Average speed in each street.

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. 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

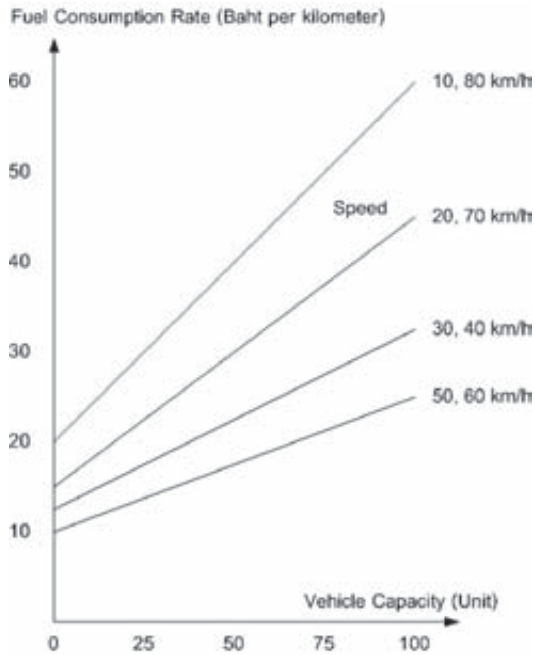

Figure 3 Fuel consumption rate depends on the average speed and vehicle capacity.

nonlinear behavior in the solution space (Kamhawi *et al*., 1996).

Part II. 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 PDTSPTC. Thus in this paper, new GA, called "Hybrid Encoding Genetic Algorithm (HEGA)" is developed for the problem of PDTSPTC.

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 PDTSPTC problem. The general structure of the HEGA can be described as follows:

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.

Decoding

When the initial population is generated, the algorithm decodes all chromosomes into sets of streets between each customer. For example,

Procedure HEGA algorithm for PDTSPTC
    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 TSPTC.

Figure 4 The procedure of HEGA

chromosome 1 in Figure 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. 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, roulette wheel 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 do 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.

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.

Reproduction operator

Reproduction is an operator that follows

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

Figure 5 Population.

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 AND DISCUSSION

In the experiment, the HEGA was implemented in Visual Basic program and tested on a labtop computer with a Celeron M 1.4 GHz processor and 756 MB RAM, running under Windows XP. The parameters, needed as input for running HEGA program, were population size (N), probability of crossover (Pc), probability of mutation (Pm) and number of generation (G). N was fixed to 50, Pc was set to 0.2, Pm was set to 0.2 (Patvichaichod and Visuwan, 2007) and G was fixed to 500. The test problem were broken up into 16 sub-problems that were varied from 5 to 8 customers. The delivery and pickup quantity of each customer was set to be varied in each sub-problem. The positive value of product quantity represented the pickup demand of customers and the negative value represented the delivery

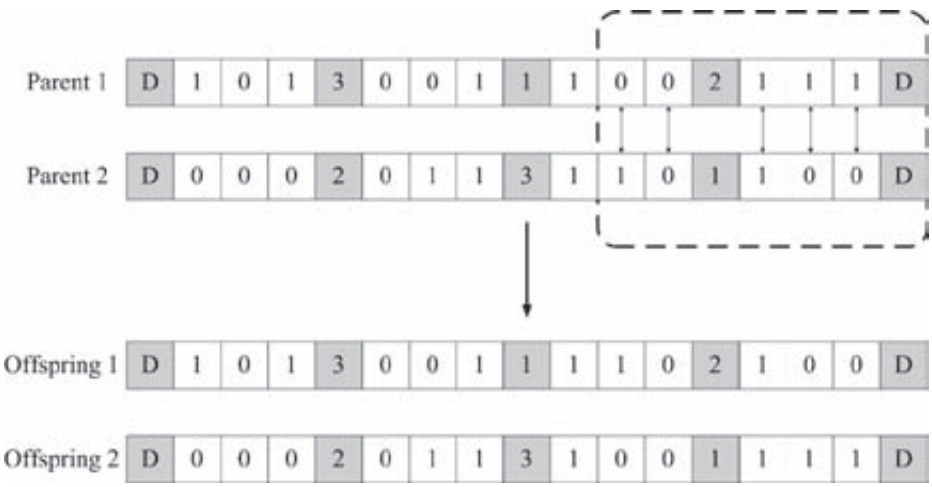

Figure 6 One-point crossover.

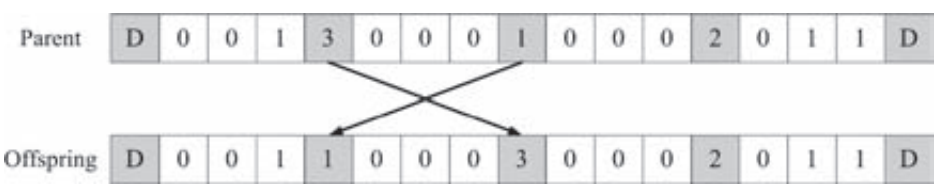

Figure 7 Exchange mutation.

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	-	-	-	14809	10840	26.8	16.9	5.6
	+10,-15	+20,-10	+5,-35	+30,-0	+0,-20	-	-	-	16371	11642	28.9	16.5	6.1
	+0,-35	+15,-0	+5,-15	+20,-10	+10,-0	-	-	-	14027	10226	27.1	17.3	5.9
	+15,-0	+0,-20	+15,-0	+0,-25	+0,-30	-	-	-	15959	11858	25.7	18.0	6.4
6	+0,-20	+25,-0	+0,-30	+10,-10	+25,-0	+10,-10	-	-	17766	12543	29.4	17.4	5.9
	+10,-15	+20,-10	+5,-35	+30,-0	+0,-20	+15,-10	-	-	17032	12433	27.0	20.1	6.4
	+0,-35	+15,-0	+5,-15	+20,-10	+10,-0	+0,-30	-	-	14369	10317	28.2	18.3	7.2
	+15,-0	+0,-20	+15,-0	+0,-25	+0,-30	+20,-0	-	-	19257	14385	25.3	15.2	7.8
7	+0,-20	+25,-0	+0,-30	+10,-10	+25,-0	+10,-10	+15,-0	-	17902	12961	27.6	19.5	6.7
	+10,-15	+20,-10	+5,-35	+30,-0	+0,-20	+15,-10	+10,-0	-	21029	15583	25.9	17.6	7.0
	+0,-35	+15,-0	+5,-15	+20,-10	+10,-0	+0,-30	+20,-10	-	15754	11548	26.7	16.2	8.1
	+15,-0	+0,-20	+15,-0	+0,-25	+0,-30	+20,-0	+10,-15	-	22330	15609	30.1	17.9	7.9
8	+0,-20	+25,-0	+0,-30	+10,-10	+25,-0	+10,-10	+15,-0	+0,-15	16687	12449	25.4	18.8	7.9
	+10,-15	+20,-10	+5,-35	+30,-0	+0,-20	+15,-10	+10,-0	+0,-10	19474	14216	27.0	19.7	8.0
	+0,-35	+15,-0	+5,-15	+20,-10	+10,-0	+0,-30	+20,-10	+25,-0	18935	13520	28.6	17.6	7.8
	+15,-0	+0,-20	+15,-0	+0,-25	+0,-30	+20,-0	+10,-15	+5,-10	20874	15071	27.8	16.0	7.6

demand. The maximum capacity of the vehicle was set to 100 units of product.

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

Travel cost saving = $(ATC_F - MTC_F) / ATC_F$

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

Convergence rate = $(MTC_B - MTC_F) / MTC_B$

Table 1 reports 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 25-30%. It is obviously seen that HEGA in the PDTSPTC test problems result in fast convergence that is about 16-20% and in all, the computational time is under 10 seconds.

CONCLUSION

This paper 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 cases. 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.

LITERATURE CITED

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

Gendreau, M., A. Hertz and G. Laporte. 1996. The traveling salesman problem with backhauls. **Computers & Operations Research** 23: 501-508.

Gendreau, M., G. Laporte and D. Vigo. 1999. Heuristics for the traveling salesman problem with pickup and delivery. **Computer & Operation Research** 26: 699-714.

Glover, F. and M. Laguna. 1997. **Tabu Search** Boston. Kluwer.

Healy, P. and R. Moll. 1995. A new extension of local search applied to the dial-a-ride problem. **European Journal of Operational Research** 83: 83-104.

Kalantari, B., A.V. Hill and S.R. Arora. 1985. An algorithm for the traveling salesman problem with pickup and delivery customers. **European Journal of Operational Research** 22: 377-386.

Kamhawi, H. N., R. S. Leclair and C. L. Philip. 1996. Feature sequencing in the rapid design system using a genetic algorithm. **J. Intell. Manuf**. 7: 55–67.

Patvichaichod, S. and P. Visuwan. 2007. Traveling salesman problem with traffic conditions, pp. 127-132. *In* **Proceeding of the 2nd International Conference on Operations and Supply Chain Management**. Bangkok.

Perez, H.H. and J.J.S. Gonzalez. 2004. A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. **Discrete Applied Mathematics** 145: 126-139.

Renaud, J., F.F. Boctor and J. Ouenniche. 2000. A heuristics for the pickup and delivery traveling salesman problem. **Computer & Operation Research** 27: 905-916.

Renaud, J., F.F. Boctor and G. Laporte. 2002. Perturbation heuristics for the pickup and delivery traveling salesman problem. **Computer & Operation Research** 29: 1129-1141.

Rosenkrantz, D.J., R.E. Strearns and P.M. Lewis II. 1997. An analysis of several heuristics for the traveling salesman problem. **SIAM Journal on Computing** 6: 563-581.

Savelsbergh, M.W.P. 1990. An efficient implementation of local search algorithms for constrained routing problems. **European Journal of Operational Research** 47: 75-85.