

Original Research Paper

Decision Support System for Open Vehicle Routing With Transshipments and Stopovers

Naragain Phumchusri and Yanipa Chinsuwan

Department of Industrial Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand

Article history

Received: 24-12-2013

Revised: 14-04-2014

Accepted: 11-08-2014

Corresponding Author:
Naragain Phumchusri
Department of Industrial
Engineering,
Faculty of Engineering,
Chulalongkorn University,
Bangkok, Thailand
Email: naragain.p@chula.ac.th

Abstract: Transshipments and stopovers are considered to be an effective method to reduce traveling distance where a transportation job can be served by two vehicles: One picks up a load and drops it at a transshipment point and then another vehicle carries that load to the final delivery place. The goal of this study is to develop a decision support system for open vehicle routing with transshipments and stopovers. We propose a heuristic to find transshipments and stopovers opportunities from an initial routing. Decision methods consist of four main processes: (1) Searching jobs that allow transshipment opportunity, (2) searching paths that allow transshipment opportunity, (3) matching paths and (4) selecting jobs to create new paths with transshipment. The output is the improved routing with transshipments and stopovers, resulting lower total costs. From computational experiments, our proposed method could reduce the system's total cost up to 12.42 percent as compared to the typical routing without transshipments and stopovers. We design system database and user interfaces, considering all input requirement entering and result displays that are easily used, so that the system can be effectively applied in actual working environments.

Keywords: Open Vehicle Routing Problem, Logistics, Transportation, Transshipments

Introduction

Logistics is widely considered as an indirect value adding activity in term of product availability in a supply chain system. Transportation is an essential activity in logistics management as it is concerned with shipping raw materials from suppliers to factories and shipping of finished goods from factories to customer locations. Logistics, especially transportation cost, was recorded to be a large portion of the majority of developing countries' economies with the increases of related factors such as fuel prices, the transportation costs of most industries continue to rise every year. Thus, transportation cost is one of the major targets to reduce in order to control the total cost of goods.

Vehicle Routing Problem (VRP) is one of the widely studied problems in logistics for the past 3-4 decades. It is concerned with determining the most suitable routes for a set of transportation demand requirements at minimum cost possible. Nag *et al.* (1988) developed

heuristics for solving variant of the vehicle routing problem where some types of vehicles are not allowed for some customer sites (certain vehicle types act as a complicating constraint). Cordeau *et al.* (2002) summarized several of the most important classical and modern heuristics for the vehicle routing problem using four criteria: Accuracy, speed, simplicity and flexibility. Laporte and Osman (1995) provided extensive literature reviews of research representing significant contributions to the field of vehicle routing. Other researches related to VRP can be found in (Taillard, 1993; Golden *et al.*, 1998; Mitrovic-Minic and Laporte, 2006; Pisinger and Ropkem, 2007; Rani and Kannan, 2014).

Open vehicle routing is considered to be an effective method to reduce transportation costs where vehicles are allowed to route without having to return back to their home base at the end of the cycle time. The goal of open vehicle routing is to reduce the total distance by eliminating empty backhauls as trucks do not have to carry empty backhauls when returning

back to the home base. This problem has received less attention and was introduced by (Sariklis and Powell, 2000) who studied open vehicle routing problem with capacity constraints using cluster first route second algorithm and applying minimum spanning tree problem. Then, (Brandão, 2004) presented a tabu search algorithm to explore the structure of this problem and discuss the situations when his method could outperform other heuristics designed for the same purpose. The examples of practical open vehicle routing problems can be found in the pickup and delivery of packages where vehicle owner contractors use their own vehicles to serve transportation requirement ordered from the center and do not return to the depot. It has been proved that Tabu search, deterministic annealing and large neighborhood search can be successfully applied to this type of problem (Li *et al.*, 2007). Other papers related to open vehicle routing problem can be found in (Tarantilis *et al.*, 2004; Fu *et al.*, 2003).

Transshipments has been introduced as an effective method to reduce traveling distance where a transportation job can be served by two vehicles: One vehicle picks up a load and drops it at a transshipment point and then another vehicle carries that load to the delivery place. Shang and Cuff (1996) proposed a multi-objective vehicle routing and scheduling heuristic for a pickup and delivery problem. Their problem included time windows, advanced requests, multi-vehicle and many-to-many transports. Thangiah *et al.* (2007) extended the work of (Shang and Cuff, 1996) and developed the design and implementation of heuristics for solving split-delivery pickup and delivery time window problems with transfer of shipments between vehicles for both static and real-time data sets. Another dynamic pickup and delivery problem with transfers was studied by (Bouros *et al.*, 2011) who proposed an algorithm to identify solutions to a request as the shortest path from a node representing the pickup location to that of the delivery location.

However, the research mentioned above did not consider opportunities of stopovers during transshipment, i.e., one vehicle needs to wait until the other vehicle arrives to transfer the load without waiting spaces at the transshipment points. In this study, we consider open vehicle routing with transshipments that allow stopovers. In our system, vehicles are allowed to route without having to return back to their home base at the end of the cycle time and the transferred load can be waiting at the transshipment points until the other vehicle arrive to pick up to the final destination. We propose a

heuristic to search for transshipments and stopovers opportunities in reducing transportation cost. The developed heuristic consists of four main step: (1) Searching jobs that allow transshipment opportunity, (2) searching paths that allow transshipment opportunity, (3) matching paths and (4) selecting jobs to create new paths with transshipment. As an effective decision making requires the incorporation of heuristic techniques into a practical system, we design a decision support system to deliver the desired output, i.e., the improved routing with transshipments and stopovers, resulting lower total transportation costs. The system database and user interfaces, taking into account of all input requirement entering and result displays, are designed so that it is easy to be applied in actual working environments.

This study is organized as follows. In the next section we explain the system structure considered in this research with heuristic concepts Section 3 shows computational results of the proposed heuristic as compared to other methods. The system database of the decision support system is described in section 4. We present the design of system's user interface consisting of input entering and result displays in section 5. Finally, section 6 concludes with a summary of insights from the results and indicates interesting future extensions of this research.

System Structure

The decision support system for continuous move routing with transshipments and stopovers consists of three major subsystems: User interface, model management and database management. Figure 1 shows the overview of the designed system architecture.

The main focus of this study is the formulation of the decision model and the development of techniques to deliver solutions of the problem. In the following subsections we discuss: (1) The definition and the formulation of the problem, (2) the heuristic concepts and (3) the computational experiments of solving the problem when using the proposed technique

Problem Description

We consider open vehicle routing with transshipments that allow stopovers. The nodes in the network represent factories, warehouses, transshipment points or customer delivery locations. The minimum-cost routes linking nodes in the network are defined for each pair of nodes. The transportation orders (or jobs) include details of products needed to be served, with origin, destination and capacity details.

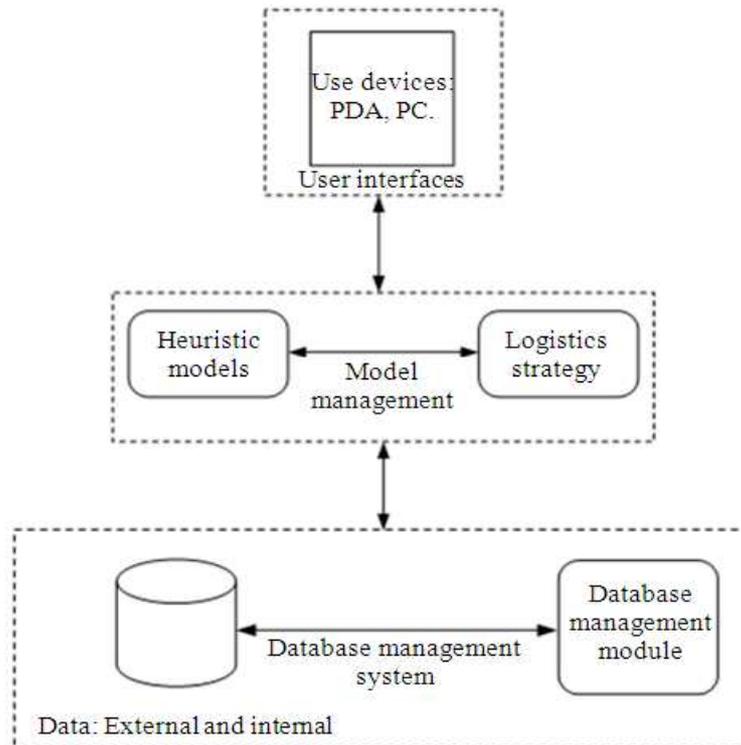


Fig. 1. The designed system architecture

The pickup and drop off time constraints are defined for each job. The initial routing of direct shipments for those jobs is available as the system input. The goal of this decision model is to seek transshipments and stopovers opportunities that provide maximum saving from the initial routing, subject to:

- Vehicle capacity constraints: The summation of weights (volume) of job carried in the same vehicle at any point of time cannot exceed the vehicle weight (volume) capacity
- Pickup and drop off constraints: Each job must be picked up at a given node and then dropped off at the other node, which are predetermined
- Delivery time window constraints: Jobs must arrive at its drop off node by the time of delivery
- Stopover constraints: Jobs can only be transshipped at the allowable transshipment nodes. At each transshipment node, loading and unloading time must be added in the transportation time. The transshipment and stopover can only be done if it does not cause any delivery delays

The decision output is the improved routing, indicating the vehicle, the pickup and drop off schedules with transshipments and stopovers locations that provide lower transportation cost, as compared to the initial routing.

Heuristic Concept

In this section, we describe each step of the designed heuristic in order to seek the transshipments and stopovers opportunity. To find the solution of this problem, we need to take into account of job constraints, truck capacity, availability of transshipment nodes and transshipment costs. The algorithm of the proposed heuristic consists of four main steps: (1) Searching jobs that allow transshipment opportunity, (2) searching paths that allow transshipment opportunity, (3) matching paths and (4) selecting jobs to create new paths with transshipments. Figure 2 summarizes the main steps of the heuristic in finding transshipments and stopovers solutions. The routing with transshipments and stopovers is determined by the following step.

Searching for Jobs that Allow Transshipment Opportunity

Jobs with transshipment opportunity can be identified as:

- Jobs that after either its pick up node or drop off nodes are removed, the total distance can be reduced
- Jobs whose visiting nodes can be alternated with other jobs, resulting lower traveling distance

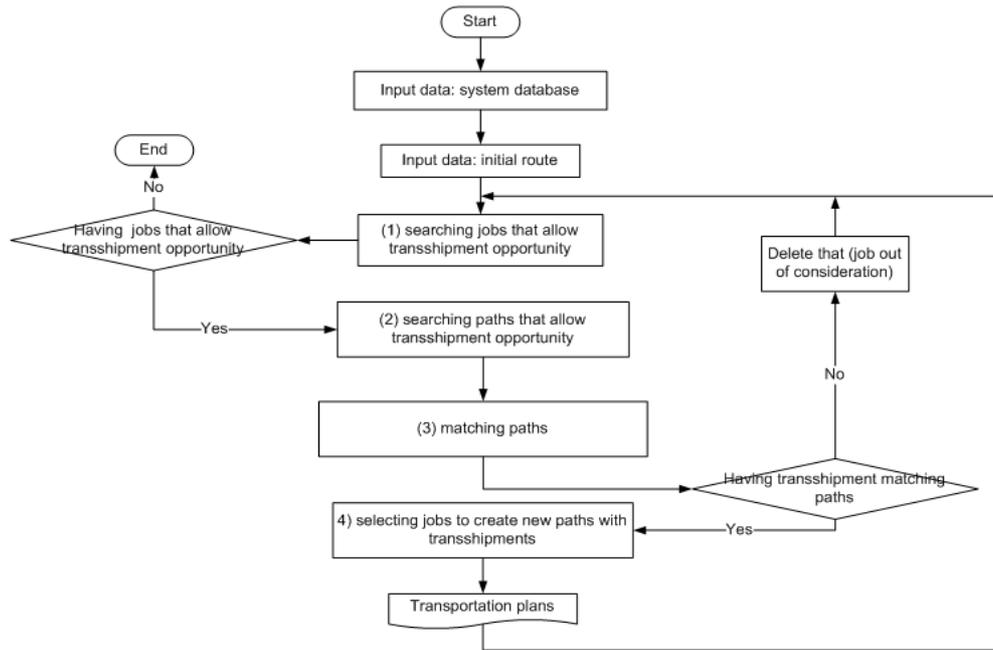


Fig. 2. Heuristic summary

Consider reduced cost from all cases and then select the one providing highest saving. Let:

$$savings_{max}^s = \text{Max}(savings_{max}^{s,1}, savings_{max}^{s,2}), \forall s \in S$$

Where:

$savings_{max}^s$ = The maximum traveling cost saved by changing paths of job s

$savings_{max}^{s,1}$ = The maximum traveling cost saved by removing either pick up or drop off nodes of job s

$savings_{max}^{s,2}$ = The maximum traveling cost saved by alternating nodes of job s with others

S = Set of all jobs

Searching Paths that Allow Transshipment Opportunity

Consider revised paths of each job in steps described in 2.2.1 and then find possible nodes to be new pick up, drop off and transshipment points, which can be selected from either the existing visiting nodes, or the nearby nodes added for transshipment. Then check the time validity of using this node. In case the node selection incurs transshipment points, this transshipment alternative will result in extra handling cost. This step will tradeoff between the saved distance and handling cost, then determine the paths that can reduce the total cost. That is:

$$oc_{ni,sv} < savings_{max}^s$$

where, $oc_{ni,sv}$ = additional cost occurred from adding transshipment nodes to the existing path to handle transportation job s with vehicle v from node I and event n .

Matching Paths

The steps in matching paths process are as followed:

- Find pairs of paths that pass conditions from the previous step and then find transshipment points for those paths. In matching, stopover time will be added to consider the possibility of transshipment. That is if a path consisting of pick up node for job s arrives at that node before the pick-up vehicle from the other path arrive, then the stopover time (or the waiting time) will be added to that particular job
- Consider the stopover time and check its validity according to the given time constraints of that job, requested from customers
- Compute cost saving from all possible transshipments for each pair of path. The maximum cost saving can be defined as:

$$\begin{aligned}
 saving_{ni,n'} &= saving_{max}^s - (OC_s \\
 &+ \sum_{i \in IM_s} \sum_{s \in S} (wating_{i,s} \times w_i) + \sum_{i \in IM_s} o_i) \\
 &\forall n \in N_i, \forall n' \in N_{i'}
 \end{aligned}$$

Where:

- saving_{ni,ni'} = Saving caused by selecting event n at node i from one path to match with event n' at node i' of the other path
- Waiting_{i,s} = Waiting time at stopover node i for transshipment of job s
- OC_s = Traveling cost increased from transshipment of job s
- w_i = Stopover cost per transportation per time occurred at transshipment node i
- o_i = Loading/unloading cost from transshipment at node i
- IM_s = Set of all possible transshipment nodes for job s
- N_i = Set of all possible event

Compute the maximum saving cost for each pair of paths for inserting transshipment of job s. That cost saving is given by:

$$\text{saving}_s = \text{Max}(\text{saving}_{ni,ni'})$$

$$\forall n \in N_i, \forall n' \in N_{i'}, \forall i \cup i' \in IM_s, \forall s \in S, i \neq i'$$

Where:

saving_s = The maximum saving cost for each pair of path after inserting transshipment of job s

Selecting Jobs to Create New Paths with Transshipment

In this step, all selected choices of transshipments for jobs obtained from the previous step are considered in the following manner:

- Select the one with maximum cost saving. That selected transshipment path are kept and used as the new input of routing
- Then repeat steps described in 2.2.2 and 2.2.3 until no more transshipments are possible for job s
- Repeat 2.2.1 to 2.2.3 until we have considered all possible job s

Figure 3 summarizes the heuristic process for all step explained above.

Results and Discussions

In this section, we perform computational experiments to assess performances of the developed heuristics. Specifically, we consider: (1) How the transportation cost savings achieved by the proposed heuristic vary at different numbers of jobs in the system (2) how the proposed

heuristic performs as compared to other methods. The coding is performed using Microsoft Visual Basic 2010 Express, with Intel Core i3, 2.2 GHz and RAM 4 Gbyte. The initial routing for the given job requirements are created with Insertion Algorithm developed in (Jaruphata and Chaovaitwongse, 2012).

In the first experiment, we test how cost savings vary when the number of jobs in the system increases. We consider three different scenarios: 20, 40 and 60 jobs. For each case, we carry out the test on 30 randomly generating instances. Table 1 displays details of instances for each set of experiment. Figure 4 shows the average percentage of transportation cost reduced after applying our heuristic to seek transshipments and stopovers opportunity, as compared to initial routing without transshipments. We found the cost saving, obtained from applying the proposed heuristic, increases when there are greater numbers of jobs in the system. Thus, a large transportation system with many required transportation jobs can achieve greater benefit from transshipments and stopovers.

In the next computational experiment, we evaluate performance in term of the reduced transportation cost achieved from the developed heuristic as compared to other works. As explained in section 2, our proposed heuristic searches for opportunities of transshipments and stopovers that can reduce transportation costs. Thus, it is interesting to explore if transshipment with stopovers can reduce cost, compared with transshipment without stopovers. In addition, we separate our heuristic into two sub-methods; allowing at most one transshipment for each job and allowing up to two transshipment for each job, in order to explore how allowing additional transshipment provide cost saving. In sum, the savings achieved from three different methods are compared.

Heuristic developed by (Kusomrata and Chaovaitwongse, 2012), considering transshipment without stopovers, i.e., do not allow stopovers at transshipment nodes.

Heuristic developed in this study. We separate our heuristic into two sub-method:

- Allowing at most one transshipment for each job
- Allowing up to two transshipments for each job

Table 2 summarizes instance details used in the experiment. Figure 5 shows the average percentage reduced traveling distance as compared to routings without transshipments. From the results, heuristic developed in (Kusomrata and Chaovaitwongse, 2012) can reduce traveling distance by 2.86, 2.95 and 1.60% for networks with 5, 10 and 15 nodes, respectively.

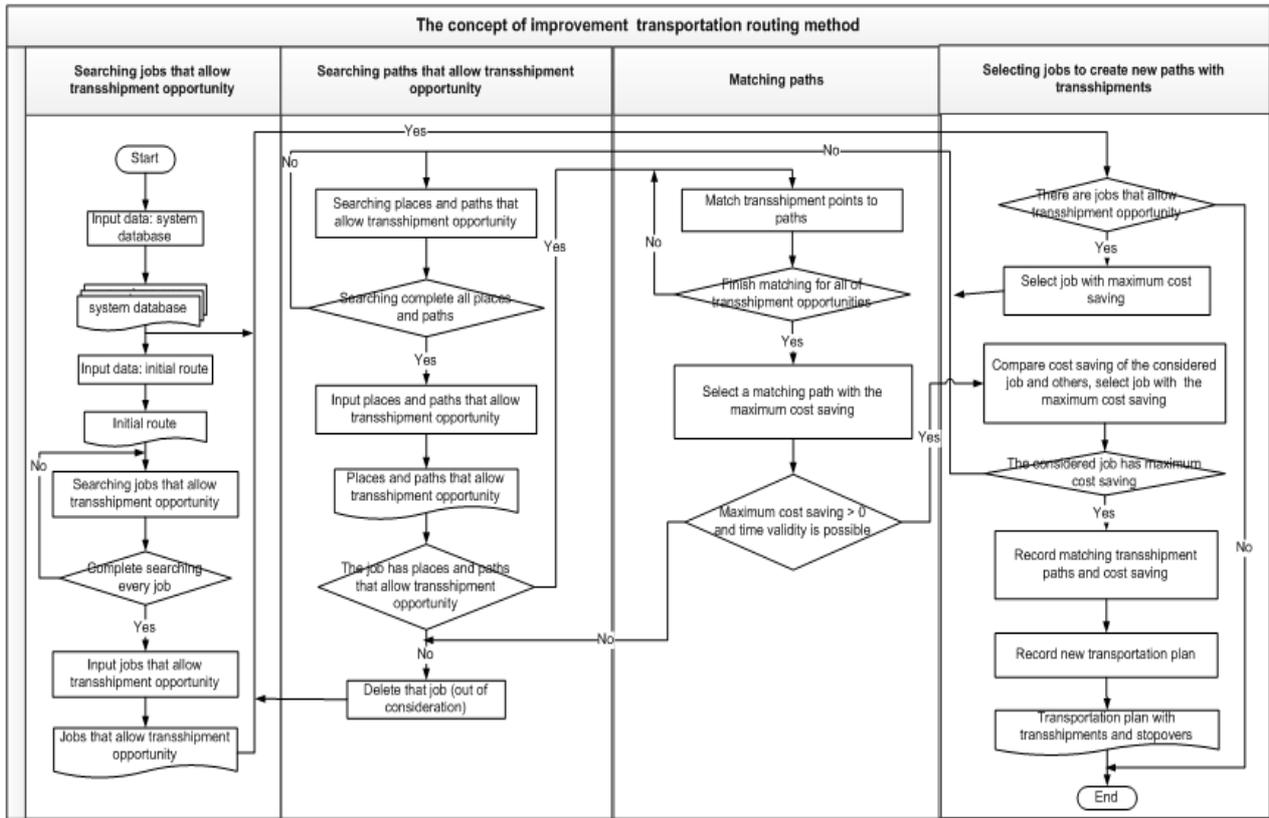


Fig. 3. Summary of the heuristics

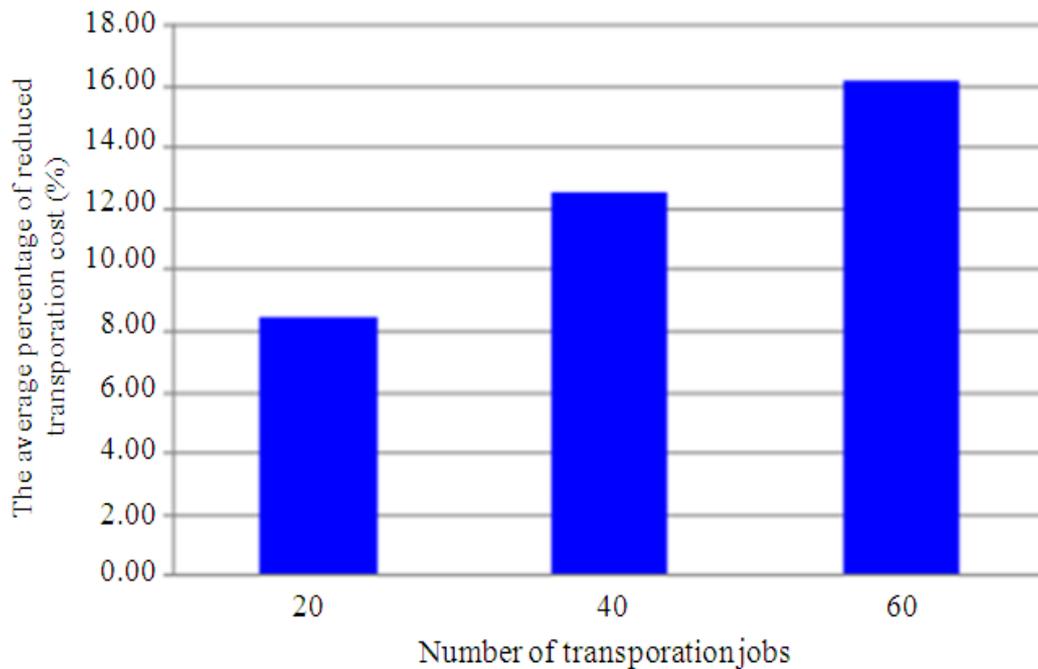


Fig. 4. The average percentage of reduced transportation cost

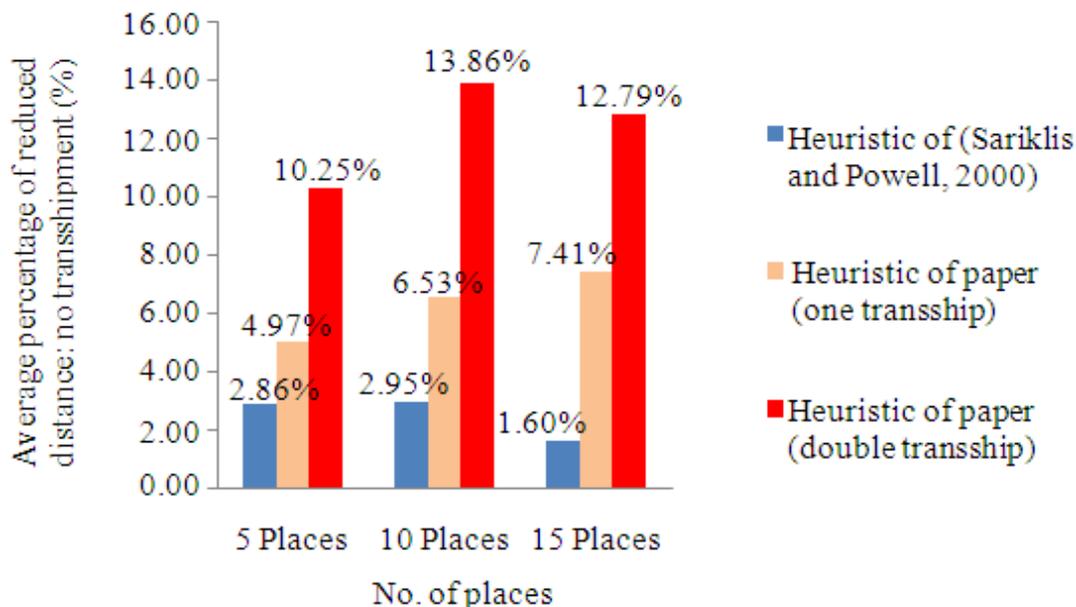


Fig. 5. The average percentage of reduced traveling distance as compared to routings without transshipments

Table 1. Details of instances for each set of experiment

Subject	Set 1	Set 2	Set 3
Number of transportation jobs	20	40	60
Number of randomly generating instances	30	30	30
Number of nodes in the network	5, 10, 15, 20, 25	5, 10, 15, 20, 25	5, 10, 15, 20, 25
Number of vehicles in the network	6	6	6

Table 2. Summary of instance details used in the experiment

Subject	Experiment details
Number of jobs	20, 40, 60
Number of experiment in each set of job	30
Number of nodes (or places) in the system	5, 10, 15
Number of vehicle	6

The heuristic developed in this study (at most one transshipment per job) can reduce traveling distance by 4.97, 6.53 and 7.41% for systems with 5, 10 and 15 nodes, respectively. When up to two transshipments are allowed per job, our heuristic provide the most distance saving at 10.25, 13.86 and 12.79% for systems with 5, 10 and 15 nodes, respectively. When comparing the overall performance of each method, the heuristic developed in this study with up to two transshipments per job provides the highest distance savings.

From the computational results, we have found the cost saving obtained from our heuristics depends on the size of transportation networks. The larger they are, the higher saving can achieve. Allowing up to two transshipments and stopovers can also provide higher cost saving. However, allowing too many transshipments

and stopover may not be preferable if the transshipment cost is large as compared to the distance saving. Thus, it is essential for the transportation manager to evaluate the number of maximum transshipments and stopovers should be allowed in each network since different numbers may be preferred in different situations.

Database

In this section, we define essential database structure for the decision support system. The system inputs are vehicle, maintenance details, places (or nodes) description and initial routing, while the output after the system processing is the improved routing with transshipments and stopovers. Input data for the system are as below:

- Vehicle input data contains the list of vehicles used for transportation in the system
- Maintenance input data contains the list of information indicating which vehicle needs to be off for maintenance, as well as time and place where the maintenance occurs
- Place (or node) input data contains the list of place where the transportation pick up, drop off and transshipment can occur in the system. They can be either factories, warehouses, transshipment points or customer delivery locations
- Distance input data contains the distance and traveling time between each pair of nodes
- Job input data contains the list of transportation jobs needed to be completed, with origin, destination and capacity details
- Initial routing data contains information of the scheduled routing to be improved by transshipment searching algorithm of the decision support system

The described input data and their details are shown in Table 3. Figure 6 present the data relationship diagram. The input data are processed according to heuristic presented in

Section 2 to determine the system output, i.e., the improved routing with lower transportation cost.

User Interface

The user interface allows interactions between users and designed system. It allows users to effectively enter the required input data and displays computed routing which is the output from the heuristic presented in section 2.

The input entering part of the user interface allows users to enter vehicles, maintenance plans, places, distances between places, jobs and initial routing. Figure 7 shows the design examples of entering screens for vehicles input data, containing the list of vehicle information, available time and location, capacity and. Figure 8 illustrates the examples of entering screens for transportation job needed to be completed, with origin, destination and capacity details. Figure 9 shows the example of entering screens for initial routing data (or jobs) to be improved for lower transportation cost. Applying our heuristic presented in Section 2, the result screen shows the improved routing with transshipments and stopovers. Figure 10 shows the example of improved vehicle routing with transshipments and stopovers results.

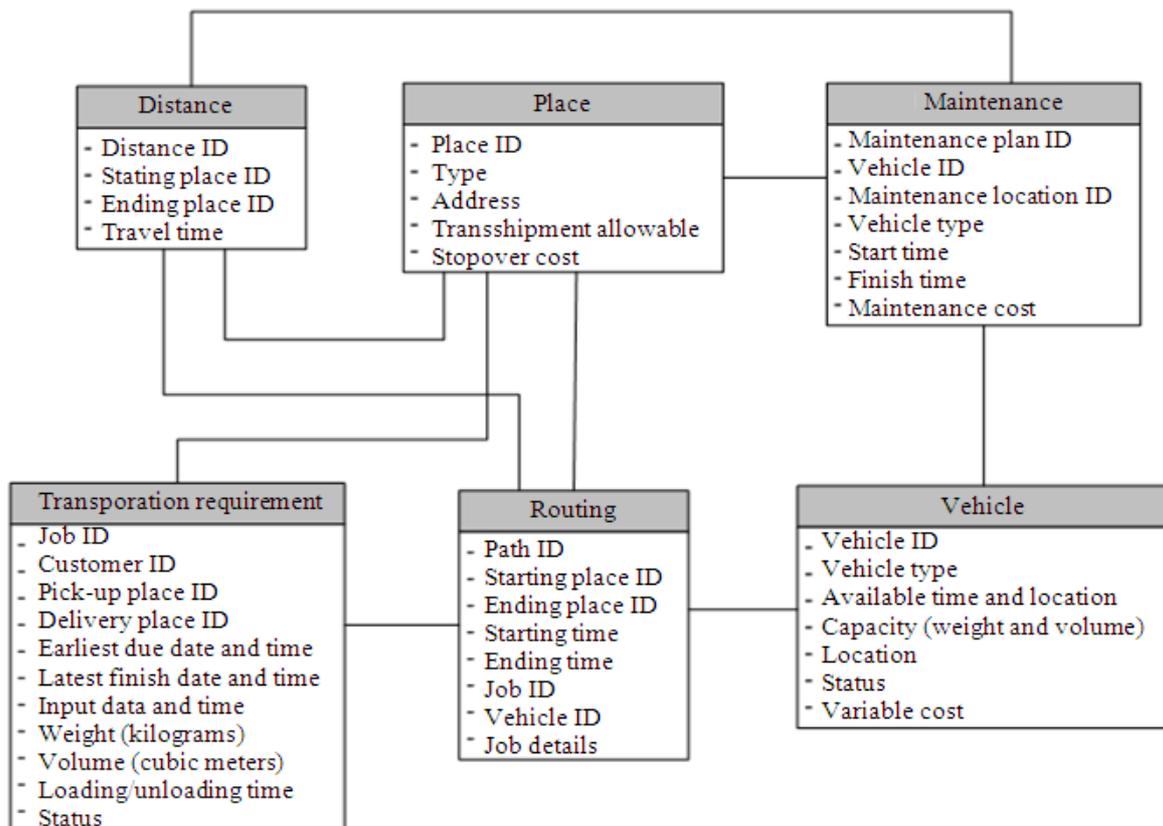


Fig. 6. Data relationship diagram

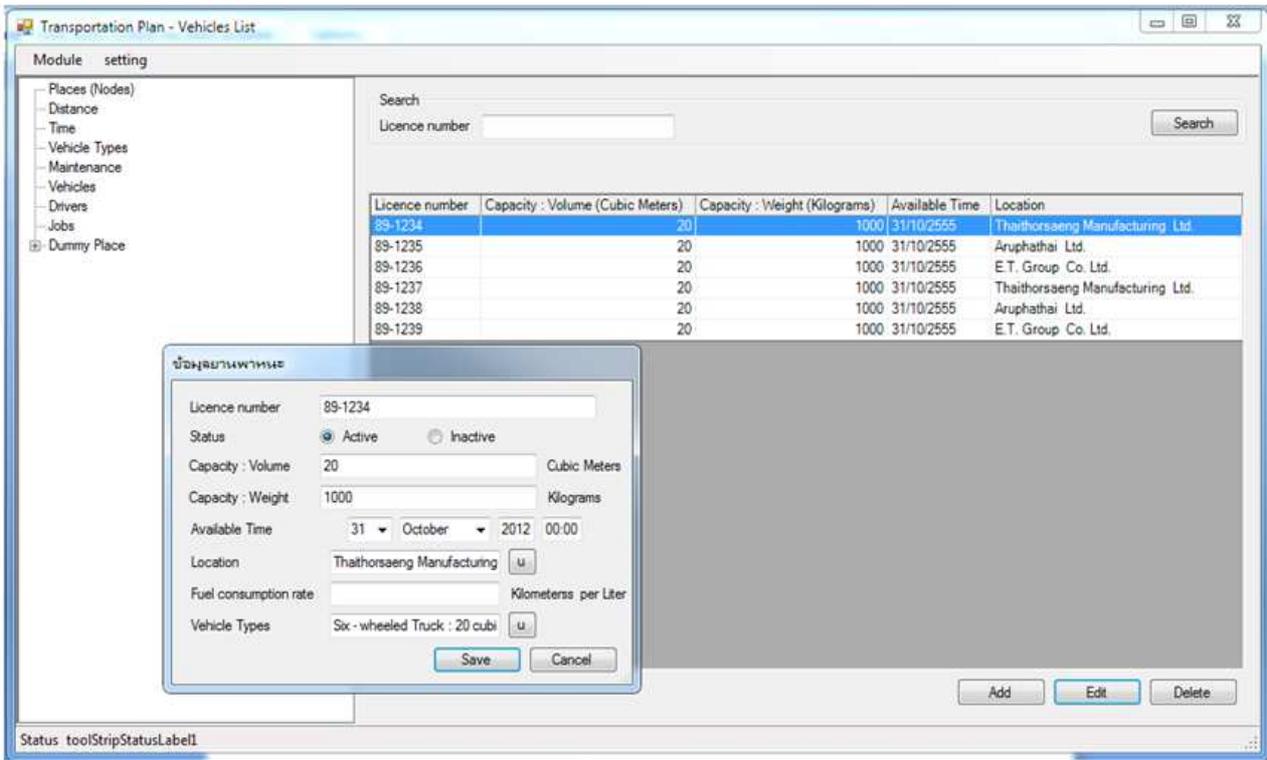


Fig. 7. Vehicle entering screen user interface

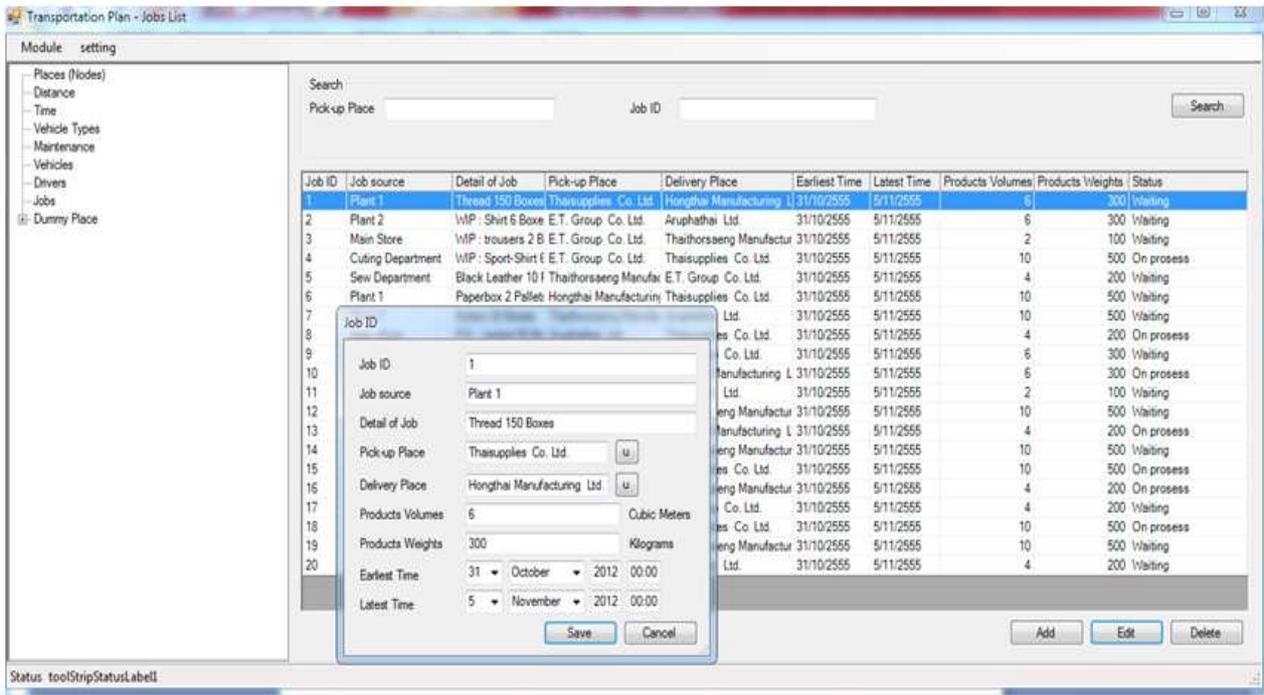


Fig. 8. Initial routing entering screen user interface

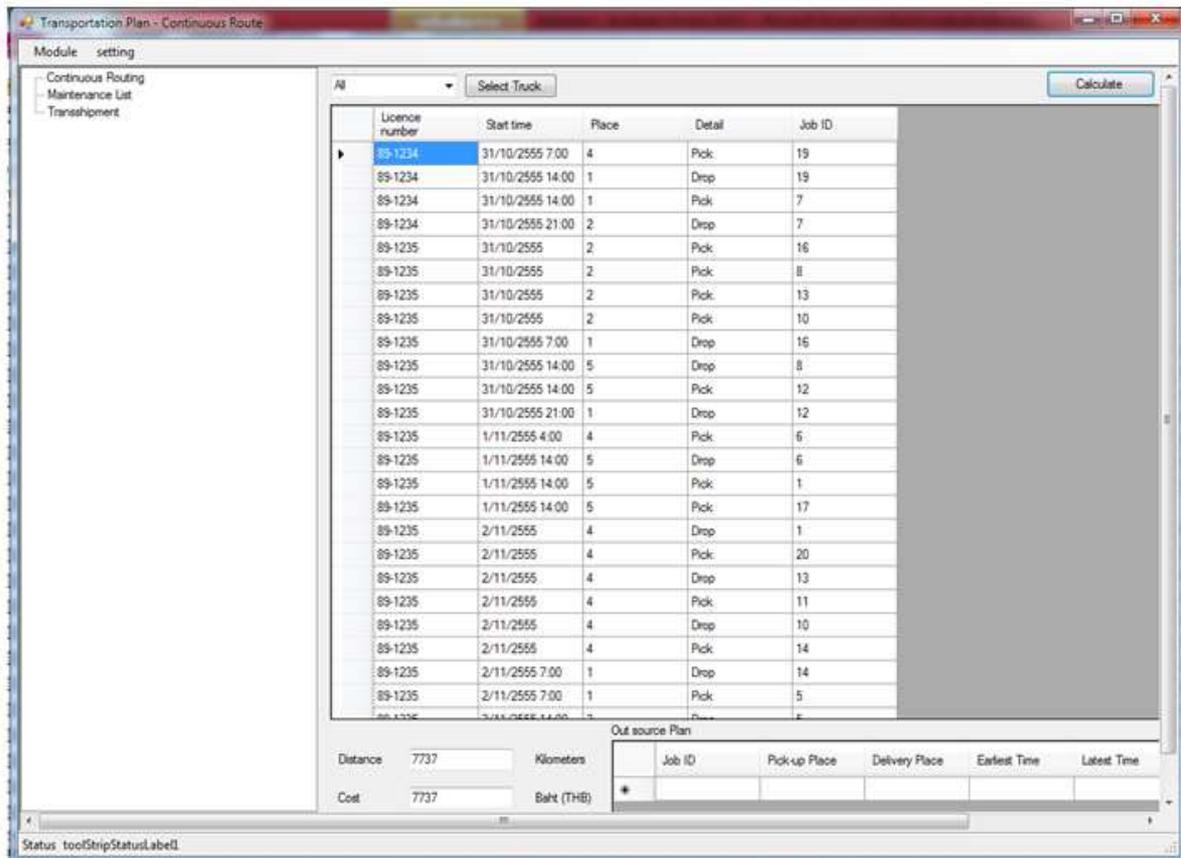


Fig. 9. Initial vehicle routing displays

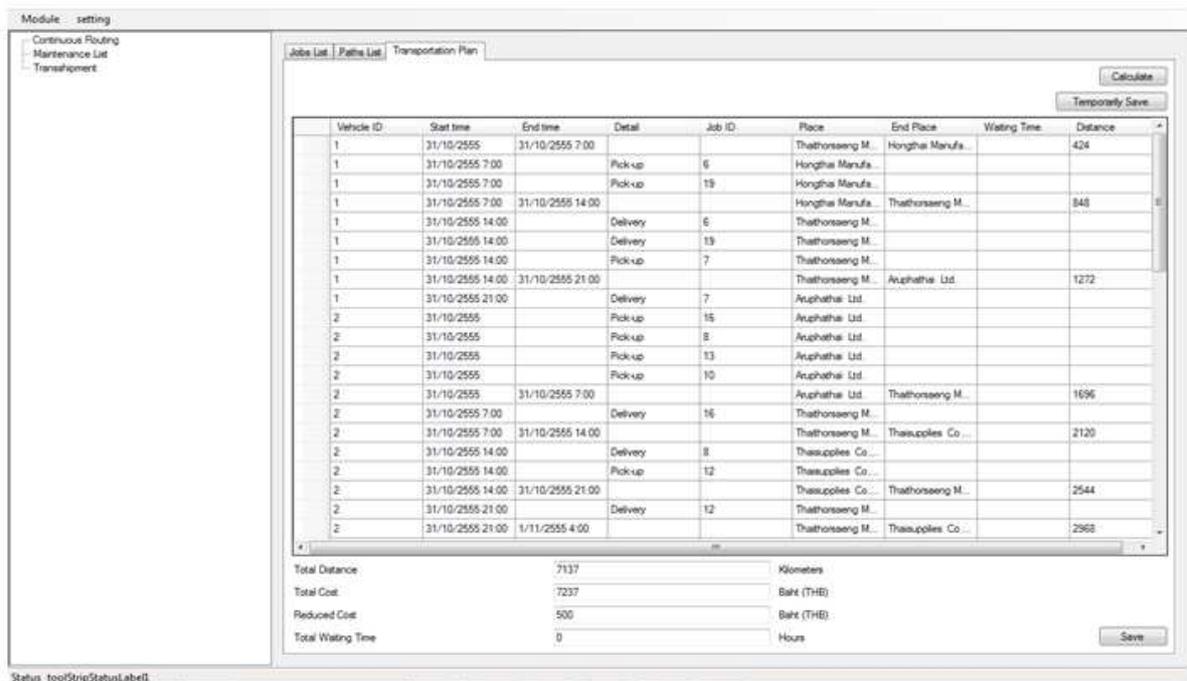


Fig. 10. Example of improved vehicle routing with transshipments and stopovers results displays

Table 3. Input data and their details classified by each

Topic	Data requirement
Vehicle	Vehicle ID Vehicle Type Available time and location Capacity (weight and volume) Status Cost/distance (Energy Consumption rate)
Maintenance	Plan Identification Number (Plan ID) Vehicle ID Vehicle Type Maintenance Location ID Start- Finish time Maintenance Cost
Place or node	Place Identification Number (Place ID) Place Type Latitude-Longitude Transshipment allowed (or not) Stopover cost (if transshipment is allowed)
Distance	Distance ID Starting Place ID Ending Place ID Travel Time (with average speed)
Job	Job ID Customer ID Pick-up place ID Delivery place ID Earliest due date and time Latest finishing date and time Input date and time Capacity: Weight Capacity: Volume Loading time Status: Waiting, processing, finished
Routing	Path ID Starting Place ID Ending Place ID Start time End time Job ID Vehicle ID Job Details: Pick-up, delivery, others

Conclusion

In this study, we developed a decision support system for open vehicle routing with transshipments and stopovers. The goal of this system is to seek transshipments and stopovers opportunities from a given routings and then create a new improved routing. The required input information are vehicle details, maintenance plans, places (or nodes) in the transportation network, distances between places, jobs requirements (e.g., transportation loads with starting and due times) and initial routings. The algorithm of the heuristic used in the system consists of four main

steps: (1) Searching jobs that allow transshipment opportunity, (2) searching paths that allow transshipment opportunity, (3) matching paths and (4) selecting jobs to create new paths. The output of the decision support system is the improved routing with transshipments and stopovers, resulting lower total transportation costs. From the computational experiments, our proposed method could reduce the system's total cost up to 12.42 percent as compared to the typical routing without transshipments and stopovers. We found that the system can achieve greater benefit from transshipments and stopovers in reducing transportation cost when there are greater

number of nodes (or places) in the network. For the method to be effectively applied in actual working environments, we designed the system database and user interfaces, taking into account of all input requirement entering and convenient result displays.

There are a number of interesting points for future research. Firstly, in our heuristic proposed in this study, there is only one job allowed in each transshipment. If more than one job are allowed to be transshipped at the same place, more savings may be possible. Secondly, we consider only one transportation mode in this study. It is interesting to extend our ideas to transshipment between different transportation modes, e.g., between trucks and trains. Thirdly, the heuristic proposed in this study can be extended and applied in more complex transportation environments such as a milk run network having many pick-up or drop-off points in one route. Of course, each transportation network has its own constraints and characters. Thus, it is essential for transportation managers to adapt our idea to best fit with their environments.

Funding Information

The authors have no support or funding to report.

Author's Contributions

All authors equally contributed in this study.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

- Bouros, P., D. Sacharidis, T. Dalamagas and T.K. Sellis, 2011. Dynamic Pickup and Delivery with Transfers. In: *Advances in Spatial and Temporal Databases*, Tao, Y., K. Mouratidis, M.A. Nascimento, M. Mokbel and Y. Huang, (Eds.), Springer Berlin Heidelberg, ISBN-10: 978-3-642-22921-3, pp: 112-129.
- Brandão, J., 2004. A tabu search algorithm for the open vehicle routing problem. *Eur. J. Operat. Res.*, 157: 552-564.
DOI: 10.1016/S0377-2217(03)00238-8
- Cordeau, J.F., M. Gendreau, G. Laporte, J.Y. Potvin and F.A. Semet, 2002. Guide to vehicle routing heuristics. *J. Operat. Res. Society*, 53: 512-522.
DOI:10.1057/palgrave/jors/260 13-19
- Fu, Z., R. Eglese and L. Li., 2003. A tabu search heuristic for the open vehicle routing problem. Technical Report 2003/042, Lancaster University Management School, Lancaster, United Kingdom.
- Golden, B., S. Raghavan and E. Wasil, 1998. *Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography*. In: *The Vehicle Routing Problem: Latest Advances and New Challenges*, Golden, B., S. Raghavan and E. Wasil, (Eds.), Springer US, ISBN-10: 978-0-387-77777-1, pp: 143-169.
- Jaruphata, O. and P. Chaovalitwongse, 2012. Heuristic for open vehicle routing problem to reduce transportation cost. *Eng. J.*, 4: 57-72.
DOI: 10.4186/ej.v4i3.231
- Kusomrata, S. and P. Chaovalitwongse, 2012. Travelling distance improvement by inter-vehicle transferring. *Eng. J.*, 4: 49-66.
DOI: 10.4186/ej.v4i1.205
- Laporte, G. and I. Osman, 1995. Routing problems: A bibliography, *Ann. Operat. Res.*, 61: 227-262.
DOI: 10.1007/BF02125421
- Li, F., B. Golden and E. Wasil, 2007. The open vehicle routing problem: Algorithms, large scale test problems and computational results. *Comput. Operat. Res.*, 34: 2918-2930.
DOI: 10.1016/j.cor.2005.11.018
- Mitrovic-Minic, S. and G. Laporte, 2006. The pickup and delivery problem with time windows and transshipment. *INFOR*, 44: 217-227.
- Nag, B., B.L. Golden and A.A. Assad, 1988. *Vehicle Routing with Site Dependencies*. *Vehicle Routing: Methods and Studies*. Studies in Management Science and Systems. Golden, B.L. and A.A. Assad, (Eds.), North Holland, Amsterdam, ISBN-10: 0-444-70407-8, pp: 149-159.
- Pisinger, D. and S. Ropke, 2007. A general heuristic for vehicle routing problems. *Comput. Operat. Res.*, 34: 2403-2435.
DOI: 10.1016/j.cor.2005.09.012
- Rani, P.K. and E. Kannan, 2014. FFA-context aware energy efficient routing using fast reactive and adaptive algorithm. *Am. J. Applied Sci.*, 11: 301-307. DOI: 10.3844/ajassp.2014.301.307
- Sariklis, D. and S. Powell, 2000. A heuristic method for the open vehicle routing problem. *J. Operat. Res. Society*, 51: 564-573.
- Shang, J.S. and C.K. Cuff, 1996. Multicriteria pickup and delivery problem with transfer opportunity. *Comput. Industrial Eng.*, 30: 631-645.
DOI: 10.1016/0360-8352(95)00181-6

Taillard, E., 1993. Parallel iterative search methods for vehicle routing problems. *Networks*, 23: 661-673. DOI: 10.1002/net.3230230804

Tarantilis, C., C. Kiranoudis and V. Vassiliadis, 2004. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Eur. J. Operat. Res.*, 152: 148-158. DOI: 10.1016/S0377-2217(02)00669-0

Thangiah, S.R., A. Fergany and S. Awan, 2007 Real-time split-delivery pickup and delivery time window problems with transfer. *Central Eur. J. Operat. Res.*, 15: 329-349. DOI: 10.1007/s10100-007-0035-x