Vehicle Capacity Planning System: A Case Study on Vehicle Routing Problem With Time Windows

Loo Hay Lee, *Member, IEEE*, Kay Chen Tan, Ke Ou, and Yoong Han Chew

***Abstract*—In this paper, we consider a local logistic company that provides transportation service for moving empty and laden containers within Singapore. Due to the limited capacity of its own fleet of vehicles, the company cannot handle all the job orders and have to outsource some orders to other smaller local transportation companies. The current operation of assigning jobs for outsourcing goes through two steps. In the first step, a certain percentage of jobs will be preselected for outsourcing according to some simple rules. Then at the second step, the rest of the jobs will be put into an in-house computer system which assigns jobs to its internal fleet of vehicles according to some greedy rules and the remaining jobs that cannot be served by the internal fleet of vehicles will be outsourced. This paper presents a vehicle capacity planning system (VCPS), which models the problem as a vehicle routing problem with time window constraints (VRPTW) and tabu search (TS) is applied to find a solution for the problem. From the simulation results, some new rules on how to assign jobs for outsourcing are derived, which are shown to be about 8% better than existing rules currently adopted by the company.**

***Index Terms*—Tabu search, vehicle capacity planning, vehicle routing problem with time windows.**

I. Introduction

STRATEGICALLY located at the crossroads of major shipping routes, Singapore port is a major port of call for about 250 shipping lines from 600 ports worldwide. In the year of 1999, 2000, and 2001, it has been rated as the second busiest container port in the world. In order to support the port activities, container related logistic services in Singapore are very prosperous. In this paper, we present a case study on a local logistic company that provides transportation services for container movement within the country. Due to the limited capacity of vehicles owned by the company, engineers in the company have to decide whether to assign the job orders of container movement to its internal fleet of vehicles or to outsource to other companies daily.

By analyzing different kinds of job orders received from the company, a general model of vehicle capacity planning system (VCPS) is built to tackle the problem. This model is similar to the vehicle routing problem with time window constraints (VRPTW). VRPTW is a problem which involves routing a fleet of vehicles, with limited capacities, from a central depot to a set of geographically dispersed customers with known demands and predefined time window constraints. The objective of VRPTW is to minimize the total travel cost with minimal number of vehicles used without violating the capacity and time constraints. It has been shown that finding a feasible solution for the traveling salesman problem (TSP), the simplest problem in vehicle routing, is a NP-complete (nondeterministic polynomial time complete) problem. Therefore, solving the VRPTW is more complicated since it involves multiple vehicles and time window constraints. Although optimal solutions to VRPTW can be obtained via exact methods, the computational time required to solve a VRPTW to optimality could be prohibitive [1] and heuristic methods are often preferred since they can find good solutions within a reasonable amount of computational time.

A great amount of work has been done in the development of heuristics for the VRPTW, which include Bodin [2], Fisher [3], Federgruen and Simchi-Levi [4], Bertsimas and Simichi-Levi [5], Laporte [6], Kohl [7], [8], Thangiah [9], Thompson [10], and Kolen [11]. These heuristics can be classified into four categories: constructive methods, route first-cluster second methods, cluster first-route second methods and incomplete optimization methods. The Saving Algorithm [12] and the heuristics of Gaskel [13], Yellow [14], and Russell [15] fall into the class of constructive methods. The optimal partitioning heuristic [16] and the Sweep Algorithm [17] belong to the class of route first-cluster second methods. For cluster first-route second methods, include the two-phase method [18], the generalized assignment heuristic [19] and the location-based heuristic [20]. The last category, incomplete optimization methods, includes cutting plane methods [21] and minimum K-tree methods [22].

Recently, heuristic search methods based on tabu search (TS) for solving the VRPTW have gained significant attention among researchers in operations research (OR) [23]–[28]. In the authors' earlier work on VRPTW [29]–[31], the tabu search method has been shown to offer good feasible solutions as compared to other competing heuristics. This paper thus focuses on developing a local search algorithm enhanced by tabu structure to tackle this vehicle routing and outsourcing assignment problem. The paper is organized as follows: Section II describes the problem. Section III describes the transportation

Manuscript received April 26, 2001; revised October 22, 2001. This paper was recommended by Associate Editor N. Pal.

L. H. Lee is with the Department of Industrial and System Engineering, National University of Singapore, Singapore (e-mail: iseleelh@nus.edu.sg).

K. C. Tan, K. Ou and Y. H. Chew are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (e-mail: eletankc@nus.edu.sg).

Digital Object Identifier 10.1109/TSMCA.2002.806498

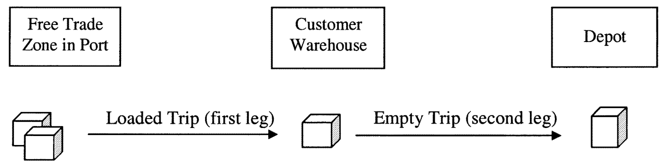


Fig. 1. Importation process of laden containers.

model and the methodology is explained in Section IV. Results are presented in Section V. A summary and a conclusion follow in Section VI.

II. Problem Scenario

A. Problems and Objectives

Everyday the company receives job orders of container movement for the next day, ranging from importation, exportation to empty container movement. The internal fleet of vehicles is used for handling these orders. However, due to the large number of job orders, most of the time, some of the job orders have to be outsourced to other companies for reasons such as exceeding fleet capacity, low revenue or urgency. The outsourcing decision is made through the following two steps.

Step 1) Manually select jobs for outsourcing by engineers according to their experience and some simple rules.

Step 2) The remaining jobs are put into an in-house computerized transportation job scheduling and monitoring system (CTJSMS) for capacity planning. A very simple rule is used in the system to assign jobs for vehicles, i.e., earliest-deadline first. With this rule, the system will pick up those jobs with earlier deadlines for their internal vehicles, until the fleet reaches its capacity limit and then the remaining jobs will be assigned for outsourcing.

Usually the capacity planning for step 2 is performed only at the end of the day when most orders have come in. Since most of the transportation companies have limited working hours, it is often impossible for the company to wait for all the orders to come before contacting the companies for outsourcing jobs. Therefore, it is important for the company to have some rules which guide them on how many jobs they should outsource and how to select those jobs for outsourcing.

The objective of this study is as follows.

1) Build a transportation model for the company and find a good solution for the problem.
2) Based on the solution obtained from the model, extract new rules on how to assign jobs for outsourcing.
3) Compare the performance of the new rules with the current rules.

B. Major Operations

There are three major types of container movement: importation, exportation and empty container movement.

1) Importation: For importation of laden containers, vehicles pick up containers at the free-trade zone area in the port and send them to customer warehouses. After discharging in the warehouses, the empty containers are sent to depots. In our model, the whole importation trip is considered as two job orders, i.e., one loaded trip from the port to a warehouse and one empty trip from the warehouse to a depot. Fig. 1 shows the importation process of laden containers.

Depending on the types of cargoes, each container has different free-storage periods at the port, for example, normal cargo has 72 hours but class two cargo (dangerous cargo) only has 24 hours of free-storage time. During this period, vehicles can go into the port at anytime to pick up the loaded containers. Meanwhile, since some of the customer warehouses and depots only operate during the usual office hours (i.e., from 8am to 6pm), this time window should also be considered in the model.

2) Exportation: Similarly, as shown in Fig. 2, for exportation, the vehicles need to pick up empty containers from depots and then send them to customer warehouses for loading. After the containers have been loaded, the company needs to book time slots at the port in order to use the crane there to move the containers when they arrive. The time slot given by the port is only 15 minutes and penalty costs are incurred when vehicles do not arrive within the time window.

The whole exportation trip will also be considered as two job orders in our model, i.e., one empty trip from a depot to a warehouse and one loaded trip from the warehouse to the port.

3) Empty Container Movement: Singapore is the empty container hub for South East Asia and a lot of shipping liners store their empty containers in the inland container depots in Singapore. Since there is a trade imbalance between different countries, from time to time, the shipping liners need to replenish their containers from one country to another. The empty container movement involves both importation and exportation. For importation activity, empty containers will be picked from the port and sent directly to depots and for exportation activity, empty containers will be sent directly to the port from depots. Usually, as opposed to other job orders, this type of job orders comes in large quantity. For instance, they might need to move 100 to 200 containers within two to three days. This process is shown in Fig. 3.

The time window for empty container movement is not strict as compared to laden container. For importation, the empty containers are taken as normal cargo and enjoy 72 hours free-storage time. For exportation, the port releases a much longer crane-booking time slot to the company, i.e., four hours per booking instead of only 15 minutes and hence the company can move many empty containers into the port at one time. This type of crane-booking is known as block booking (BB).

C. Job Details

In general, when a company receives a job order, it includes the following information:

1) Job type (importation, exportation or empty container movement);
2) Laden/empty trip;
3) Normal cargo/class 2 cargo;
4) Trailer type (20 or 40 feet);
5) Source/destination location
6) Handling time in source/destination location;
7) Time windows for source/destination location.

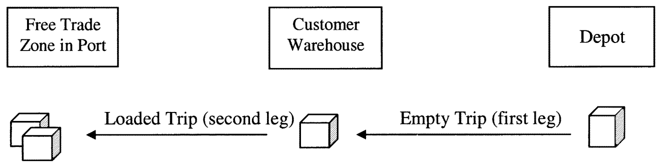


Fig. 2. Exportation process of laden containers.

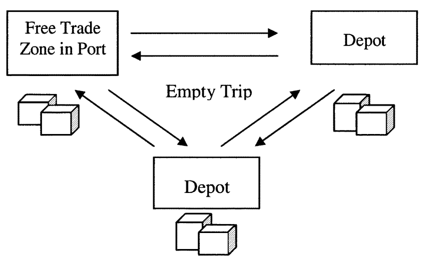


Fig. 3. Empty container movement.

Time window information of each job is important as it determines the feasibility of the job scheduling. To determine the time windows, we need to know the details of the vessel information, such as estimated arrival time (ETA), estimated departure time (ETD), complete of discharge (COD), ready time, latest time, and crane booking slot. The time windows vary significantly from type to type, for example, normal cargo importation jobs enjoy time windows of 72 hours at the port, but exportation jobs only have 15 minutes crane booking time slot at the port.

In this study, we divide all the job orders into seven types, ranging from the importation, exportation to empty container movement. These job orders are listed below and illustrated in Fig. 4.

T1) Importation of normal cargo from port to warehouse.
T2) Importation of class 2 cargo port to warehouse.
T3) Exportation of normal cargo from warehouse to port.
T4) Exportation of class 2 cargo from warehouse to port.
T5) Importation of empty containers from port to depot.
T6) Exportation of empty containers from depot to port.
T7) Empty container movement from warehouse to depot or depot to warehouse.

From Fig. 4, it can be noticed that T1 to T4 are laden trips, while T5 to T7 are empty trips.

III. Transportation Model

A. Generalized Job Model

The generalized model of a job order can be described in Fig. 5.

To process a job order, we first need to travel to the source location of the order with a trailer. Since there is time window constraint in the source location, we might need to wait until the time window is reached and then the agent at the source location (which can be either the port, warehouses or depots) will handle the container and load it to the trailer. Once the container is picked up, it will be sent to the destination location and the respective agent at the destination location will receive and process the container.

There are two types of containers with two different lengths: 20 feet and 40 feet. Before the trucks go to pick up a container in the source location, it needs to travel to the nearest trailer exchange point to collect the correct type of trailer. Since the company holds a lot of trailers and the ratio of trucks to trailers can be as high as 1:9, we can always assume that the right type of trailers is always available at every trailer exchange point. In other words, the number of job orders will never exceed the trailer capacity and hence, the trailer type feasibility constraints are not considered in the model. With the knowledge of the location for trailer exchange point, we can always factor in the traveling time to and from the trailer exchange location into the computation of the traveling time from the job starting point to the source location. Although the trailer type does not affect the feasibility of designing a specific route, it contributes to the overall routing performance because the costs of handling different types of containers are different.

Under this job model, the vehicle routing and outsourcing assignment problem to be tackled is transformed into a VRPTW with slight modifications.

B. Mathematical Model

As shown in Fig. 6, the VRPTW problem in this paper consists of a set of identical vehicles, a set of customer job orders represented by nodes and a network connecting the vehicles and job orders. It is assumed that there are N job orders and K vehicles. Each arc in the network represents a connection between two jobs and indicates the job handling sequence. Each route starts from a truck set-off point, followed by the job orders handled by this truck. The number of routes in the network is equal to the number of vehicles used and one vehicle is dedicated to one route. Notice that this network does not represent the real geographical connection between job locations.

Each job order in the network can be visited only once by one of the vehicles. The time window constraints imposed by each job must be satisfied. Vehicles are also required to complete their individual route within a preset maximal route time, as the drivers have fixed working hours. From the above description, it can be noticed that there are several major differences between our vehicle routing model and the VRPTW model proposed by Solomon [32].

1) Our model does not have a central depot. vehicles can start and end the routes at any location on the map. In the conventional VRPTW model, there is a central depot and all the routes must start and end at this depot.
2) In our model, each node represents a job order and connections between nodes indicate the job handling sequence. In VRPTW model, each node represents the geographical location of a customer demand point.
3) Only one container can be moved at a time. In the conventional VRPTW model, each vehicle has a limited capacity and can carry several loads provided that it is not overloaded.
4) Jobs in our model are more complicated than those in conventional VRPTW model. As defined in Section III-A,

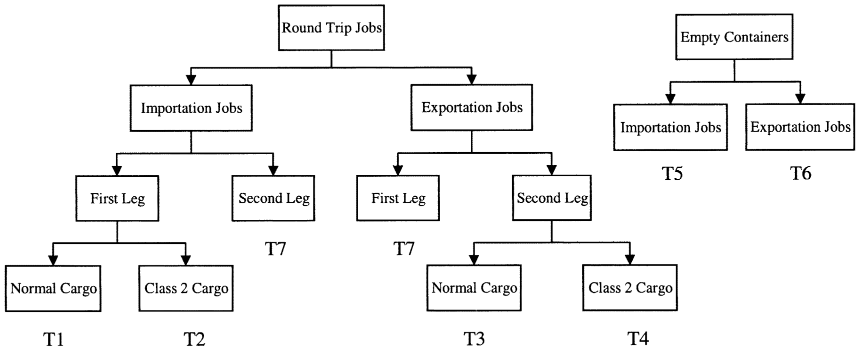


Fig. 4. Job orders distribution.

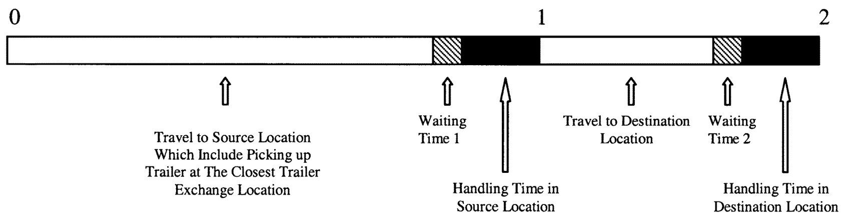


Fig. 5. Time sequence of generalized job model.

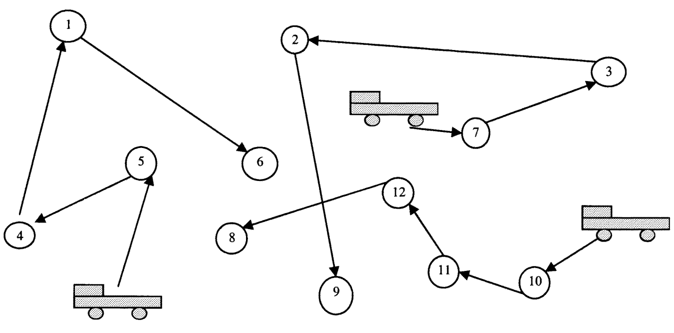


Fig. 6. Vehicle routing model.

each job in our model must cover two locations: source and destination respectively. In contrast, jobs in VRPTW have only one service location per single job.

5) The objective in our problem is to minimize total cost instead of minimizing total distance traveled.

The variables and parameters used in the model are defined as follows:

Decision Variables: $X_{ikm} \in \{0, 1\}$. $i \in \{1, \ldots, N\}$; $k \in \{1, \ldots, K\}$; $m \in \{1, \ldots, M\}$. If job i is assigned to truck k as the mth job, $X_{ikm} = 1$, otherwise 0.

$X_{i0} \in \{0, 1\}$. $i \in \{1, \ldots, N\}$. If job i is outsourced to companies (subscript 0 represents other transportation companies), $X_{i0} = 1$, otherwise, 0.

$W_{mk1} (m \in \{1, \ldots, M\}; k \in \{1, \ldots, K\})$ is the waiting time in source location for the mth job of truck k.

$W_{mk2} (m \in \{1, \ldots, M\}; k \in \{1, \ldots, K\})$ is the waiting time in destination location for the mth job of truck k.

Parameters:

N	Number of jobs.
K	Number of tracks.
M	Maximal number of jobs that can be handled by one truck in a day.
P_i	Cost of job i, if job i is handled by internal truck, where $i \in \{1, \ldots, N\}$.
S_i	Cost of job i, if job i is outsourced, where $i \in \{1, \ldots, N\}$.
T_{mk0}	Starting time of the mth job of truck k;
T_{mk1}	Starting time to go to destination location of the mth job of truck k.
T_{mk2}	Ending time of the mth job of truck k.
D_{ij}	Travelling time from the destination of the ith job to the source of the jth job.
D_i	Travelling time between source and destination location for the ith job.
H_{i1}	Handling time in source location for the ith job.
H_{i2}	Handling time in destination location for the ith job.
R_{i0}	Starting time of the source time window for the ith job.
R_{i1}	Ending time of the source time window for the ith job.
R_{i2}	Starting time of the destination time window for the ith job.
R_{i3}	Ending time of the destination time window for the ith job.
A_{k0}	Starting time of the available time period for truck k.
A_{k1}	Ending time of the available time period for truck k.

The time sequence of the mth job for truck k is represented in Fig. 7: See (1)–(9) at the bottom of the next page.

Equation (1) is the objective function that represents the total cost. For the same job order, P_i is usually less than S_i, which means that the company can save more money if it handles the job itself rather than by outsourcing. Our vehicle capacity planning system is designed to select the so-called "right jobs" for the company's own fleet in order to minimize the total cost. Equation (2) defines that every job can only be assigned once. Equation (3) requires the jobs to be assigned to trucks sequen-

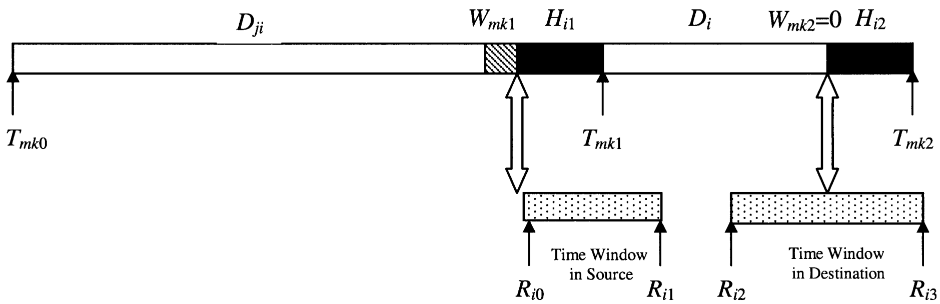


Fig. 7. Time sequence of mathematical model.

tially. Equation (4)–(6) are the time sequence for each job. Equation (7) requires that all jobs are finished within the truck's available working hours. Equations (8) and (9) are the time window constraints for the source and destination locations.

IV. Methodology

In recent years, a great amount of work has been done on the development of heuristics for the VRPTW problem. Among these methods, TS has been shown to achieve significant improvement in optimizing the solutions. In this paper, we adopt TS based on λ-interchanges as the method for solving the VRPTW problem. TS is powerful in searching for solution neighborhood [24] and [27], as compared to other heuristics which may get stuck in local minima. The algorithm is described in details in the following sections.

A. Initial Solution and λ-Interchange Local Search Method

We assume that there are a total of K trucks (or K routes) and a job pool with all available job orders. To generate the initial solution, we randomly select job orders and insert them sequentially into each route by using the standard push-forward insertion method. The push-forward insertion method will only

$$\text{Minimize} \quad \sum_{i=1}^{N} P_i \left(\sum_{k=1}^{K} \sum_{m=1}^{M} X_{ikm} \right) + \sum_{i=1}^{N} S_i X_{i0} \tag{1}$$

$$\text{Subject to} \quad \sum_{k=1}^{K} \sum_{m=1}^{M} X_{ikm} + X_{i0} = 1, \text{ for } i \in \{1, \ldots, N\}. \tag{2}$$

$$\sum_{i=1}^{N} X_{ik(m+1)} \le \sum_{i=1}^{N} X_{ikm} \le 1,$$
$$\text{for } k \in \{1, \ldots, K\} \text{ and } m \in \{1, \ldots, M-1\}. \tag{3}$$

$$T_{(m+1)k0} = T_{mk2}, \text{ for } k \in \{1, \ldots, K\} \text{ and } m \in \{1, \ldots, M-1\}. \tag{4}$$

$$T_{mk1} = T_{mk0} + \sum_{\substack{i=1, j=1 \\ i \ne j}}^{N} X_{ikm} X_{jk(m-1)} \left(D_{ji} + H_{i1} \right) + W_{mk1}$$
$$\text{for } k \in \{1, \ldots, K\} \text{ and } m \in \{1, \ldots, M\}. \tag{5}$$

$$T_{mk2} = T_{mk1} + \sum_{i=1}^{N} X_{ikm} \left(D_i + H_{i2} \right) + W_{mk2}$$
$$\text{for } k \in \{1, \ldots, K\} \text{ and } m \in \{1, \ldots, M\}. \tag{6}$$

$$A_{k0} \le T_{mk0} \le A_{k1} - \left(T_{mk2} - T_{mk0} \right),$$
$$\text{for } k \in \{1, \ldots, K\} \text{ and } m \in \{1, \ldots, M\}. \tag{7}$$

$$R_{i0} \le \sum_{k=1}^{K} \sum_{m=1}^{M} X_{ikm} \left(T_{mk1} - H_{i1} \right) + X_{i0} R_{i0} \le R_{i1}$$
$$\text{for } i \in \{1, \ldots, N\}. \tag{8}$$

$$R_{i2} \le \sum_{k=1}^{K} \sum_{m=1}^{M} X_{ikm} \left(T_{mk2} - H_{i2} \right) + X_{i0} R_{i2} \le R_{i3},$$
$$\text{for } i \in \{1, \ldots, N\}. \tag{9}$$

allow a job order to be inserted at the place where the feasibility of the route can be maintained. If the job cannot be inserted into the current route, it will be put into a new route. The procedure will continue until no job order can be inserted in any of the K route. All the unassigned job orders are then assigned to truck 0 (or route 0), which represents the subcontractors. Note that there are no time window constraints for this "truck 0." We have a total of $K+1$ routes in the initial solution and the total cost of this solution is calculated accordingly. The following is one of the possible solutions.

$$\begin{array}{ll} \text{Route } 1: & 2 \rightarrow 12 \rightarrow 6 \rightarrow 11 \rightarrow 7 \\ \text{Route } 2: & 15 \rightarrow 3 \rightarrow 14 \rightarrow 8 \rightarrow 1 \rightarrow 10 \\ \quad \cdots\cdots & \\ \text{Route } 0: & 4, 5, 9, 13, \ldots \end{array}$$

From route 1 to route K, the solution indicates the job assignments and handling sequence. Route 0 contains the outsourced jobs. After getting the initial solution, λ-interchange local search method is used to generate the neighborhood structure. The λ-interchange local search method was introduced by Osman [26] to improve the solution by interchanging jobs between sets of routes. Based on some successfully solved problems [14], [26], and [33], the λ-interchange local search method has been shown to be an effective neighborhood-searching algorithm.

The local search procedure is conducted by interchanging jobs between routes. For a chosen pair of routes, the searching order for the jobs to be interchanged needs to be defined, either systematically or randomly. Here, we consider the cases of $\lambda = 1$ and $\lambda = 2$, which means that at most two jobs may be interchanged between routes. Based on the value of λ, there are a total of eight interchange operators: (0,1), (1,0), (1,1), (0,2), (2,0), (2,1), (1,2), and (2,2). The operator (1,2) on a route pair (R_p, R_q) indicates a shift of two jobs from R_q to R_p and a shift of one job from R_p to R_q. The other operators are defined similarly. For a given operator, the jobs are considered sequentially along the routes. These operators define the neighborhood of a solution. The approach is to search through the neighborhood and rank all possible moves according to their evaluation. Subsequently the first ranked move will be adopted unless it is in the tabu list.

B. Tabu Search

TS uses memory structures to support and encourage a non-monotonic search [34]. TS stores the most recent moves or visited solutions in a tabu list. Attempts that reverse the moves or reproduce the solutions in the tabu list will be marked as "tabu" and be denied. However, an aspiration criterion can release this restriction if a move leads to a new global best solution. The lifetime of a tabu status in the tabu list is controlled by the tabu list size, where first-in-first-out rule is often used for refreshing the tabu list.

There are usually two kinds of structures for tabu list. The first one records the recent moves of individual jobs. The structure is described below

$$\{R_1, \text{Job1}; \quad R_2, \text{Job2}; \quad R_3, \text{Job3}; \cdots\cdots\}. \tag{10}$$

A pair, $\{R_i, \text{Job}i\}$ indicates that Job$i$ is moved out of route R_i. Any move later which attempts to shift Job$i$ back to its original route R_i is prohibited as long as $\{R_i, \text{Job}i\}$ is still in the tabu list.

The other structure records the whole route information, e.g.,

$$\text{Route } 1: \quad 2 \rightarrow 12 \rightarrow 6 \rightarrow 11 \rightarrow 7.$$

If any of the jobs in this route is removed, the whole route will be recorded as "tabu." The elements of this structure are strings of job numbers representing recently visited routes.

$$\{2-12-6-11-7;\ 15-3-14-8-1-10;\ \cdots\cdots\}. \tag{11}$$

The tabu restriction imposed by this structure is similar to the first structure. Any future move will be prohibited if it attempts to produce the same route that has been encountered before. Compared with the first structure, the second structure provides a more detailed and accurate information of the recent moves, although it takes longer computational time for comparison. The second structure is adopted in this work and the size of the tabu list is set to ten.

C. The Heuristic

After defining the tabu structure and the local search method, a heuristic is proposed to solve the problem. At the start of the heuristic, an initial solution is generated as described in Section IV-A and then the λ-Interchange Local Search Method is applied to explore the neighborhood of this initial solution. During the search, route 0 will be paired with each route from route 1 to route K and the λ-interchange operators will examine all the possible moves between each pair of routes that can result in feasible new solutions. The total cost of these newly generated solutions is calculated and put into a candidate list in ascending order.

The move that is ranked first in the candidate list will be checked for validity, i.e., whether it is a "tabu" or not. If it is not Tabu, this move will be adopted and the solution it produces will be set as the new current solution. After refreshing the tabu list, this iteration is completed. If the first ranked move is tabu, then the second ranked candidate will be checked until a legal move is found. The whole process is described as follows:

Heuristic: Tabu Search
TABU-1: Obtain an initial feasible solution S and set $S_b = S$ (S_b is the current best solution).
TABU-2: Initialize the Tabu List and Candidate list to the empty set.
TABU-3: Set the iteration counter $m = 0$.
TABU-4: Explore the neighborhood of solution S using the method of λ-Interchange Local Search. Update the candidate list in ascending order: the best move is placed in first position.
TABU-5: Set S' be the first candidate in the candidate list.
TABU-6: If $\{\text{Cost}(S') < \text{Cost}(S_b)\}$
  Go to TABU-8.

```
TABU-7: If {S' is Tabu}
   Select the next best candidate in the can-
didate list as S' and go to TABU-6.
TABU-8: Update S = S' and Tabu list.
  If {Cost(S) < Cost(S_b)}
   Set S_b = S.
  m = m + 1.
TABU-9: If {m is greater than a preset number}
   Terminate the program.
  Else
   Go to TABU-4.
```

There is an aspiration condition in TABU-6: if the candidate move can produce a global best solution, the tabu status of this move is overruled. Because of the use of TS, the above heuristic has the ability to overcome the problem of being trapped in a local optimum. When the current solution falls into a local optimum and there are no other moves which lead to a better solution, the heuristic has to select a "worse" move which results in deterioration. By allowing it to adopt a "worse" move and preventing any repetition or cycling, TS forces the current solution to find a way out of the local optimum to search for a better solution.

V. Results Analysis

A. Experiment Scenario

The company provided us with historic archive data for analyzing the results of our transportation model. The general information includes the average total number of job orders the company receives every month, the order distribution and other necessary job information such as the distribution of time windows. Based on the information, we randomly generated 14 data sets. Each data set contains one month's job orders together with all the job details. These 14 data sets are divided into two groups, the first group contains seven data sets which is used for **rule extracting** and the second group contains the other seven data sets for the purpose of **rule testing**. The VCPS model built in Sections III and IV is applied to solve the problem with the proposed tabu search heuristic.

The experiment goes through three phases:

1) apply VCPS to solve the 7 rule-extracting data sets;
2) by analyzing the results from 1), extract new rules on how to outsource jobs;
3) apply both the new rules and current rules to solve the seven rule-testing data sets and compare their performance.

B. VCPS Results

In this section, the VCPS is applied to solve one of the seven rule-extracting data sets and the best results obtained at each iteration are shown in Fig. 8. Notice that the costs of the solutions have been normalized with the cost of the final solution obtained. As can be seen, the algorithm is quite effective in improving the solution during the initial phase of the optimization. However, for latter phase of optimization, it has to spend more time to explore the neighborhood in order to escape from local optima. Although there is no improvement in the best result obtained from iteration nine to iteration ten, the algorithm is not trapped at local optima and is able to find better solution in the later iteration, due to the integrated tabu search mechanism in the algorithm.

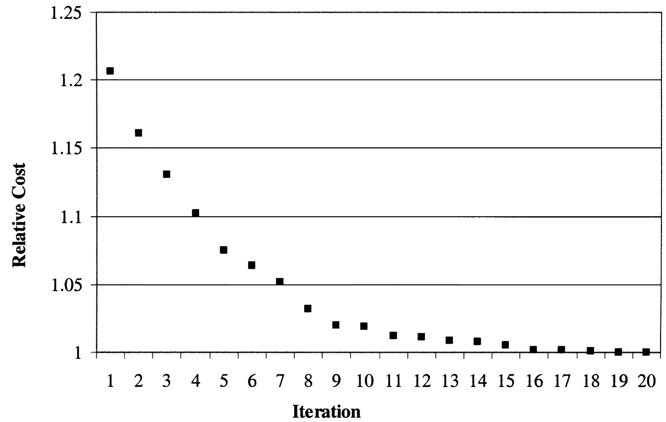


Fig. 8. Results of the VCPS.

On the other hand, we also compare the Tabu Search Algorithm with the pure λ-Interchange Local Search Algorithm in solving the 7 rule-extracting data sets. The results of the comparison are shown in Table I. The first column of Table I indicates the 7 test cases that we are solving in the problem and second column shows the costs of the initial solutions generated in each test cases. The next two columns show the costs of the best solutions obtained by the pure λ-Interchange Local Search Algorithm and the Tabu Search Algorithm respectively. Notice that all the costs have been normalized by the best costs found in each instance. From the results, we have found that Tabu Search Algorithm has consistently obtained better results than the λ-Interchange Local Search Algorithm.

C. New Rules Extraction From VCPS Results

In this section, we applied the VCPS to solve the vehicle routing problem for each of the 7 rule-extracting data sets. Since all job orders are considered in the VCPS, the Tabu Search heuristic is capable of obtaining solutions not only on how to assign jobs to individual trucks, but also on which jobs should be assigned for outsourcing. Table II summarizes the average percentage of the jobs that are assigned for outsourcing for each job type.

The first column of Table II indicates the job types with different trailer sizes and the second column indicates the percentage of jobs being outsourced for each job type. The next two columns show the relation between the percentage of outsourcing and the job types with different job distances. The standard deviations of the results are about 0.7%. From the table, we can observe some important information.

1) This table considers different job types, trailer types and job distances. In Table II, we have divided jobs into 14 categories, i.e., T1/20, …, T7/40. Each category is further divided into two sub-groups, i.e., short distance group and long distance group. As a result, there are 28 different types of job orders.
2) From the percentage of outsourcing, we can identify which types of jobs are more suitable for outsourcing. From the table, we can observe that some types of jobs have been outsourced completely (e.g., T5/20) while some types of job only use the internal fleet (e.g., T1/40).

TABLE I
COMPARISON OF ALGORITHM WITH λ-INTERCHANGE LOCAL SEARCH ALGORITHM

Test case	Initial Solution	Local Search	Tabu Search
1	1.2064	1.1599	1.0000
2	1.2132	1.1315	1.0000
3	1.2375	1.1633	1.0000
4	1.2072	1.1310	1.0000
5	1.2226	1.1201	1.0000
6	1.2325	1.1905	1.0000
7	1.2090	1.1701	1.0000

TABLE II
RESULTS WHEN ONLY VCPS IS APPLIED

		Short Distance	Long Distance
	% of Outsourcing	% of Outsourcing	% of Outsourcing
T1 / 20	9	0.64	20.9
T1 / 40	0	0	0
T2 / 20	64	25	92.3
T2 / 40	26.2	2.8	53.3
T3 / 20	64.2	51.1	82
T3 / 40	41.7	25.1	62.8
T4 / 20	74.1	70	85.7
T4 / 40	67.9	57.9	88.9
T5 / 20	100	100	100
T5 / 40	50.7	26.8	83.3
T6 / 20	100	100	100
T6 / 40	80.8	70.2	96.2
T7 / 20	100	100	100
T7 / 40	70.4	39.1	85.9

Following the results from Table II, we have proposed two different rules here:

New Rule A:

1) New rule considers job type, trailer type, and distance factor.
2) According to Table I, all job orders have been divided into 28 different job types with different outsourcing priorities.
3) Higher priority jobs are chosen first for outsourcing. For example, T5/20 with short distance (having a outsourcing priority of 100%) will have a higher priority than T2/40 with long distance (having a outsourcing priority of 53.3%).

New Rule B:

1) This set of new rule is a simplified version of new rule A, where the distance factor is ignored and only the job type and trailer type are considered.
2) The 28 job types in new rule A are aggregated into 4 groups and the priorities of outsourcing are calculated respectively. The outsourcing priority for each group (G1, G2, G3, and G4), is obtained by finding the average of the outsourcing percentage of all the jobs in the group:
   (G1) All empty jobs (T5, T6, and T7) are combined together. From the results in Table I, empty jobs have a very high percentage of outsourcing, which is close to 79.2%.
   (G2) All the 20 ft trailer jobs in T2, T3, and T4 are combined and the outsourcing percentage is about 65%.
   (G3) All the 40 ft trailer jobs in T2, T3, and T4 are combined and the outsourcing percentage is 41.4%.
   (G4) All T1 jobs are considered as one group and the outsourcing percentage is 3%.

As a result, G1 has the highest priority for outsourcing, followed by G2, G3, and G4.

D. Comparison of New Rules and Current Rules

The strategy that is currently adopted in the company is quite simple and contains two main rules.

1) 50% of empty container movements will be outsourced.
2) Those jobs requiring long distance travel will be outsourced.

In order to compare the performance of the new rules and current rules for each set of the testing data, we first select the jobs for outsourcing using the priority given by each rule. The number of jobs that are selected for outsourcing will be decided by a parameter called "preselect percentage."After the jobs have been selected for outsourcing, the rest of the jobs will be put into VCPS, which will then decide how to assign and sequence jobs for the internal fleet as well as for outsourcing. It is obvious that if the preselect percentage is low, the results obtained will be close to the solutions obtained without applying any rules. However, the results will be worse if the preselect percentage is high, since a lot of jobs that should not be outsourced during the preselect phase will be selected. The performances of the current and proposed new rules are compared in Table III.

The "percentage of outsourcing" is the total percentage of outsourcing including the jobs that are outsourced in both preselect and VCPS steps. The "cost" ($) is the total cost representing the final objective value. The cost value is normalized with respect to the best solution found without applying any pre-select rules. All the figures in this table are the averages taken over the 7 rule-testing data sets and the standard deviation is about 0.7%. These results are plotted graphically for easy comparison as shown in Fig. 9.

1) From the figure, it can be observed that when the preselect level is below 40%, the total cost is almost at the same level for all the three sets of rules, e.g., when the preselect level is low, the second stage of VCPS still has a lot of room to optimize the results and hence the solutions obtained are close to the best result without any pre-select rules.
2) When the preselect level increases, the total cost also increases. From the figure, we can see that New Rule B gives the best results among the three.
3) From Table II, we can see that the capacity limit of the fleet is about 35% of the total jobs generated. It is interesting to see that if we set our preselect level at around 65%, New Rule B gives a better result than the other two rules and its cost is only about 2% above the best results obtained without any preselect action. Therefore, it is suggested to set the preselect level at 65%.

TABLE III
COMPARISON AMONG RULES

	New Rule A		New Rule B		Current Rules	
Pre-Select %	% of Outsourcing	Cost($)	% of Outsourcing	Cost($)	% of Outsourcing	Cost ($)
21	56.2	1.000454	60.3	1.007235	59.7	1.006441
24	57.5	1.000711	60.8	1.010636	59.1	1.000982
27	57.5	1.001317	60.1	1.003825	60	1.001978
30	58	1.002276	61	1.005112	60.2	1.010101
33	57.6	1.002936	61.4	1.008066	60.5	1.013405
36	57.1	1.003662	61.3	1.005136	60.9	1.003762
39	57.5	1.004457	61.9	1.00451	61.4	1.015915
42	57.6	1.009049	62.4	1.00402	62.1	1.010324
45	58	1.012723	63	1.006281	62.1	1.013697
48	58	1.018333	63.8	1.006425	62.1	1.028853
51	58.3	1.026731	62	1.006051	62.2	1.038429
54	58.6	1.034648	63	1.010316	62.5	1.060858
57	59.2	1.048197	62.2	1.014361	63.3	1.07576
60	61	1.059625	62.5	1.020509	64	1.101269
63	63.5	1.072795	63.6	1.03649	65.7	1.122647
66	66.2	1.095792	66.2	1.054912	67.9	1.15413
69	69.2	1.113305	69	1.083514	69.6	1.183685
72	72	1.130397	72	1.114426	73.2	1.220922
75	75	1.153001	75	1.151495	75.2	1.254971
78	78	1.195293	78	1.192347	78.6	1.298203
81	81	1.232374	81	1.229188	81.7	1.327229

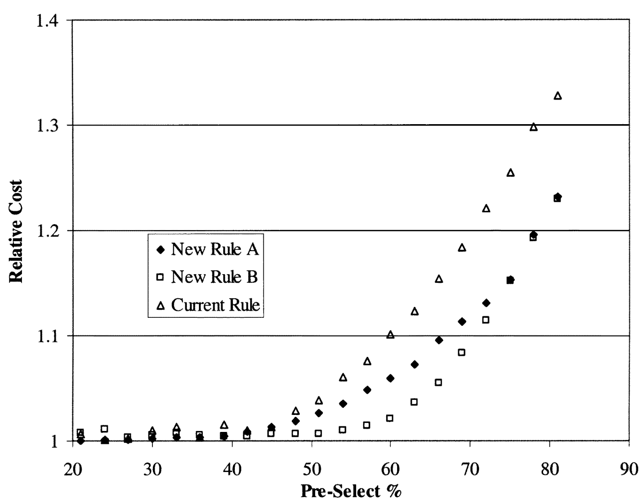


Fig. 9. Rules comparison.

4) For the average cost saving within the capacity limit range of 60–66%, New Rule A can save 4.75% and New Rule B saves 8.14%, as compared to the current rules.

VI. Conclusions

In this paper, a transportation model for container movement has been built to solve the outsourcing problem faced by a transportation company. Because of the large amount of job orders, the company must select some jobs to outsource and the proposed VCPS has helped to select jobs and to minimize the total cost. The transportation model has been built with mathematical definitions and the advanced artificial intelligence method of tabu search heuristic has been chosen to solve the problem, after careful examination of various heuristics studied in the authors' previous works [29]–[31]. Based on the raw data information provided by the company, performance of the current system and the proposed VCPS has been compared. It has been shown that the new rules extracted from the proposed VCPS can save the total cost up to 8.14% as compared to the current system adopted by the company.

Acknowledgment

The authors would like to thank The Logistics Institute—Asia Pacific (TLI-AP) and the company for granting the invaluable chance of this industrial collaboration project. They also wish to thank the editors and anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the quality of this paper.

References

[1] M. Desrochers, J. Desrosiers, and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Oper. Res.*, vol. 40, no. 2, pp. 342–354, 1992.

[2] L. Bodin, B. Golden, A. Assad, and M. Ball, "The state of the art in the routing and scheduling of vehicles and crews," *Comput. Oper. Res.*, vol. 10, no. 2, pp. 63–211, 1983.

[3] M. L. Fisher, "Vehicle routing," in *Handbooks Opererations Research and Management Science*, M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, Eds. Amsterdam, North-Holland, 1995, Network Routing, pp. 1–33.

[4] A. Federgruen and D. Simchi-Levi, "Analytical analysis of vehicle routing and inventory routing problems," in *Handbooks Operations Research and Management Science*, M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, Eds. Amsterdam, North-Holland, 1995, Network Routing, pp. 297–373.

[5] D. Bertsimas and D. Simchi-Levi, "The new generation of vehicle routing research: Robust algorithms addressing uncertainty," *Oper. Res.*, vol. 44, pp. 286–304, 1996.

[6] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *Eur. J. Oper. Res.*, vol. 59, pp. 345–358, 1992.

[7] K. Niklas, J. Desrosiers, O. B. G. Madsen, M. M. Solomon, and F. Soumis, "2-path cuts for the vehicle routing problem with time windows," *Trans. Sci.*, vol. 33, no. 1, pp. 101–116, 1999.

[8] K. Niklas and O. B. G. Madsen, "An optimization algorithm for the vehicle routing problem with time windows based on lagrangean relaxation," *Oper. Res.*, vol. 45, no. 3, pp. 395–406, 1997.

[9] S. R. Thangiah, I. H. Osman, R. Vinayagaoorthy, and T. Sun, "Algorithms for vehicle routing problems with time deadlines," *Amer. J. Math. Manage. Sci.*, vol. 13, no. 3&4, pp. 323–355, 1994.

[10] P. M. Thompson and H. Psaraftis, "Cyclic transfer algorithms for multi-vehicle routing and scheduling problems," *Oper. Res.*, vol. 41, pp. 935–946, 1993.

[11] A. W. J. Kolen, A. H. G. Rinnooy Kaan, and H. W. J. M. Trienekens, "Vehicle routing with time windows," *Oper. Res.*, vol. 35, no. 2, pp. 266–273, 1983.

[12] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Oper. Res.*, vol. 12, pp. 568–581, 1964.

[13] T. J. Gaskel, "Bases for vehicle fleet scheduling," *Oper. Res.*, vol. 18, pp. 281–295, 1967.

[14] P. Yellow, "A computational modification to the saving method of vehicle scheduling," *Oper. Res.*, vol. 21, pp. 281–283, 1970.

[15] R. A. Russel, "An effective heuristics for the M-Tour Traveling Salesman Problem with some side constraints," *Oper. Res.*, vol. 25, pp. 521–524, 1977.

[16] J. Beasley, "Route first-cluster second methods for vehicle routing," *Omega 11*, pp. 403–408, 1983.

[17] B. E. Gillett and L. R. Miller, "A heuristic algorithm for the vehicle routing dispatch problem," *Oper. Res.*, vol. 22, pp. 340–349, 1974.

[18] N. Christofides, A. Mingozi, and P. Toth, "The vehicle routing problem," in *Combinatorial Optimization*. New York, 1978, pp. 318–338.

[19] M. L. Fisher and R. Jaikumar, "A generalized assignment heuristic for vehicle routing," *Networks*, vol. 11, pp. 109–124, 1981.

[20] J. Bramel and D. Simchi-Levi, "A location based heuristic for general routing problems," *Oper. Res.*, vol. 43, pp. 649–660, 1995.

[21] G. Cornuéjols and F. Harche, "Polyhedral study of the capacitated vehicle routing problem," *Math. Program.*, vol. 60, pp. 21–52, 1993.

[22] M. L. Fisher, "Optimal solution of vehicle routing problems using Minimum K-Tree," *Oper. Res.*, vol. 42, pp. 626–642, 1994.

[23] G. Bruno-Laurent, J. Y. Potvin, and J. M. Rousseau, "A parallel implementation of the Tabu search heuristic for vehicle routing problems with time window constraints," *Comput. Oper. Res.*, vol. 21, no. 9, pp. 1025–1033, 1994.

[24] W. C. Chiang and R. A. Russell, "A reactive Tabu search metaheuristic for the vehicle routing problem with time windows," *INFORMS J. Comput.*, vol. 9, pp. 417–430, 1997.

[25] P. Jean-Yves, T. Kervaut, B. L. Garcia, and J. Rousseau, "The vehicle routing problem with time windows—part I: Tabu search," *INFORMS J. Comput.*, vol. 8, no. 2, pp. 158–164, 1996.

[26] I. H. Osman, "Metastrategy simulated annealing and Tabu search algorithms for the vehicle routing problems," *Ann. Oper. Res.*, vol. 42, pp. 421–451, 1993.

[27] E. D. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Y. Potvin, "A Tabu search heuristics for the vehicle routing problem with soft time windows," *Transp. Sci.*, vol. 31, pp. 170–186, 1997.

[28] S. R. Thangiah, "A hybrid genetic algorithm, simulated annealing and Tabu search heuristics for the vehicle routing problem with time windows," in *Practical Handbook of Genetic Algorithms: Complex Coding Systems*, L. Chambers, Ed. Boca Raton, FL: CRC, 1999, vol. III, pp. 253–277.

[29] K. C. Tan, L. H. Lee, Q. L. Zhu, and K. Ou, "Heuristic methods for vehicle routing problem with time windows," *Artif. Intell. Eng.*, vol. 15, no. 3, pp. 281–295, 2001.

[30] K. C. Tan, L. H. Lee, and K. Ou, "Hybrid genetic algorithms in solving vehicle routing problems with time window constraints," *Asia-Pac. J. Oper. Res.*, vol. 18, pp. 121–130, 2001.

[31] ——, "Artificial intelligence techniques in solving vehicle routing problems with time window constraints," Eng. Applicat. Artif. Intell., to be published.

[32] M. M. Solomon, "Algorithms for vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–365, 1987.

[33] I. H. Osman and N. Christofides, "Capacitated clustering problem by hybrid simulated annealing and Tabu search," *Int. Trans. Oper. Res.*, vol. 1, no. 3, pp. 317–336, 1994.

[34] F. Glover, "Tabu search-part I," *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, 1989.

Loo Hay Lee (M'97) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C, in 1992 and the S.M and Ph.D. degrees in 1994 and 1997, respectively from Harvard University, Cambridge, MA.

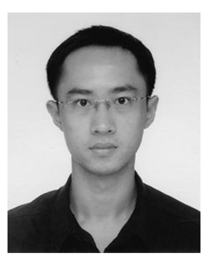
He has been an Assistant Professor in the Department of Industrial and Systems Engineering, National University of Singapore, Singapore since 1997. His research interests include simulation-based optimization, production scheduling and sequencing, logistics and supply chain planning, as well as vehicle routing.

Dr. Lee is a member of ORSS and INFORMs.

Kay Chen Tan received the B. Eng. degree with first class honors in electronics and electrical engineering, and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He was with the Center for Systems and Control and the Evolutionary Computing Group, Glasgow, U.K., before joining the Department of Electrical and Computer Engineering, National University of Singapore, as an Assistant Professor in 1997. He has authored or coauthored more than 80 technical publications and has served as a program committee or organizing member for many international conferences. He is currently an Associate Editor for the *International Journal of Systems Science*, Taylor and Francis, London. His current research interests include computational intelligence, evolutionary computing, intelligent control, and engineering designs optimization.

Dr. Tan is Associate Editor for IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.



Ke Ou received the B.Eng. and M.Eng. degree in electrical and computer engineering from the National University of Singapore in 1998 and 2001 respectively.

He is currently a software engineer in Singapore Technologies. His research interests include evolutionary optimization and vehicle routing.

Yoong Han Chew received the B.Eng. degree with first class honors in computer engineering from the University of Technology, Malaysia, Johor Bahru, in 2001. He is currently pursuing the M.Eng. degree at the National University of Singapore.

His current research interests include evolutionary multiobjective optimization, artificial intelligence, constraints handling in solving vehicle routing problems.