Endüstri Mühendisliği Dergisi
Cilt: 29 Sayı: 3-4 Sayfa: (31-41)

Özgün
Makale

A CONSTRUCTION MATHEURISTIC FOR MULTI-TRIP VEHICLE ROUTING PROBLEM AT SANTA FE-INDONESIA

Jeroen Wouter De Boer, Bahar Çavdar, Haldun Süral*

Department of Industrial Engineering, METU
bcavdar@metu.edu.tr, sural@ie.metu.edu.tr

Geliş Tarihi: 19.10.2017; Kabul Ediliş Tarihi: 12.12.2018

ABSTRACT

This study is motivated by a real-life application of the multi-trip vehicle routing problem (VRPM). The VRPM relaxes a strong assumption that each vehicle can perform only a single trip. Even though this problem setting is more suitable for many applications, the literature on the VRPM is limited compared to the other variants of the Vehicle Routing Problem. In this paper, we propose a construction matheuristic based on a set covering approach, and provide the results of computational experiments for a more involved variant of VRPM.

Keywords: Vehicle routing, matheuristic, real-life application

SANTA FE-ENDONEZYA İÇİN ÇOK-SEFERLİ ARAÇ ROTALAMA PROBLEMİ İÇİN KURUCU BİR MAT-SEZGİSEL

ÖZ

Bu çalışma, çok-seferli araç rotalama probleminin (VRPM) gerçek hayattaki bir uygulaması ile motive edilmiştir. VRPM her aracın sadece tek bir yolculuk yapabileceği varsayımını gevşetir. Bu problem tipi birçok uygulama için daha uygun olsa da, VRPM ile ilgili literatür Araç Rotalama Probleminin diğer çeşitleriyle karşılaştırıldığında sınırlıdır. Bu çalışmada, belirli bir küme kaplama yaklaşımına dayanan kurucu bir mat-sezgisel yöntemi geliştirilmiş ve daha kapsamlı bir VRPM varyantı için sayısal deneylerin sonuçları sunulmuştur.

Anahtar Kelimeler: Araç rotalama, mat-sezgisel, gerçek hayat uygulaması

* İletişim yazarı

1. INTRODUCTION

The vehicle routing problem (VRP) and its variants have been vastly studied in the literature. Most of the studies assume that vehicles will be assigned to a single route. However, when the number of vehicles is rather limited as it is observed in the small-sized logistics companies or the vehicles have small capacities because of the traffic regulations, this assumption does not hold. So, we may need to assign multiple routes to a vehicle while determining the minimum-cost routes.

There are many applications where the multi-trip VRP (VRPM) is suitable. Distribution problems in city logistics are of that kind since generally only small-capacity vehicles are allowed in urban areas by laws and regulations. For example, the routing problem for online grocery shopping is a specific application of the VRPM gaining popularity especially in cities that has to use small vehicles to meet such regulations and customer requirements in a timely manner.

Despite the number of applications which can benefit from the VRPM setting, this problem has not received as much attention as the other variants of the problem in the literature. Taillard et al. [13] propose a three-step tabu search heuristic. They assume a fixed number of vehicles with a limit on the planning period duration. Brandao and Mercer [3] compare the results of their tabu search algorithm with Taillard et al. [13]. Although they do not provide the details on the solution quality, they report that their approach outperforms the competitors on the infeasible instances. Salhi and Petch [11] propose a multi-step heuristic that iteratively generates a diverse set of routes to increase the chance of finding better solutions. Olivera and Viera [8] develop an algorithm based on adaptive memory programming principle. Yang and Tang [14] introduce a filter and fan approach to the problem. They first create an initial solution using a tabu search based on insertion, swapping, and 2-opt, then improve the solution using a dynamic local search procedure. Mingozzi et al. [7] present an exact method which is based on set-partitioning. Cattaruzza et al. [4] develop a hybrid genetic algorithm for the problem. They adapt the split procedure developed by Prins [9] and introduce a new local search operator.

The studies integrating the time-windows to the problem are much more recent. Azi et al. [1] propose the frst exact solution method for the VRPM with time-windows (VRPMTW). They use column generation embedded in a branch and price algorithm. Hernandez et al. [6] provide a two-phase exact algorithm for the VRPMTW. They enumerate the routes subject to the maximum allowable duration in the frst phase and choose the best set of routes by a branch and price approach in the second phase. Azi et al. [2] propose an adaptive large neighborhood search. The underlying mechanism is to destroy a part of the current solution and reconstruct it to obtain a better solution.

Although these methods in the literature are very competitive in terms of the solution quality, it is still challenging to solve large instances in reasonable times. Moreover, the problems in practice often pose scenarios that are considerably more complicated than the ones studied in the literature. For instance, to the best of our knowledge, there is no study in the literature on the VRPM with time windows and pickup and delivery. Motivated by a real-life problem provided by a distribution company in Jakarta which runs a milk run system with their heterogeneous feet, in this paper, we present a set covering-based construction matheuristic for an inclusive variant of the VRPM. Since many express delivery companies run a milk run system with a large feet of vehicles as high utilization rate is one of the main goals, the insights drawn from the case study may apply to the express delivery industry.

The rest of the paper is organized as follows. In Section 2, we describe Santa Fe case study in detail. In Section 3, we present our set covering-based construction matheuristic. In Section 4, we propose new methods to create routes and present the computational results. Finally, we present our final remarks in Section 5.

2. DESCRIPTION OF THE CASE STUDY

Santa Fe Indonesia is a company that offers relocation services to individuals as well as companies. They have a single warehouse in South Jakarta and heterogeneous feet to perform three different types of transportation, namely inbound shipments, outbound shipments and local moves.

Inbound shipments arrive at the (air)port of Jakarta and once released from customs, they are brought to the warehouse. These shipments are then prepared to be transported to their final destinations in Jakarta (or beyond). Most shipments arrive in containers at the port of Jakarta. After custom procedures are completed, shipments are released and transported to the warehouse. Goods are transferred to liftvans and delivered to the final destination upon request of the customer. Outbound shipments are packed into liftvans at the site of origin, and then transported to the warehouse. Here the shipment is prepared for air or sea-freight transportation. The last type of transport is local moves; both the origin and the destination are located within the neighborhood of Jakarta office. These shipments may be transported to the destination directly or via the warehouse.

Although, Santa Fe is serving a small number of customers, the problem they face is quite inclusive. Inbound shipments must be handled at a predetermined time interval after custom operations are done. Similarly, outbound shipments are scheduled for specific vessels, so they need to arrive at the airport accordingly. The vehicle fleet consists of three types of vehicles, $K=\{1,\ldots, k^1, k^1+1,\ldots, k^2, k^2+1,\ldots, k^3\}$, where k^p is the number of vehicles of type p. The vehicle distinction is based on the capacity differences. The customer set is partitioned into two: N_1 is the set of customers with positive demand (delivery) and N_2 is the set of customers with negative demand (pick-up). The company generally allows a route to have either only delivery or pickup customers. This is motivated by the fact that many companies in the express delivery industry use milk run systems as helps to decrease the inventory holding cost, balance the utilization and work-load of the vehicles, and increase the speed of transportation.

Most shipments involve household goods including appliances, consumer electronics and other fragile items. Valuable items need to be placed into custom-made wooden crates before transportation. So, intense labor is required at the origin and the destination site. For this reason, a joint crew and routing approach would be suitable since an item could not be transported before it is carefully packed. However, in this case study we do not take the crew assignment into consideration because temporary crew members can be hired at all times with small cost if needed. We consider regular crew working hours a day, though.

In Santa Fe case, the main reason behind assigning multi-trips to the vehicles is the capacity restriction. Having to connect the routes to arriving and departing fights is another reason since it is not preferable or feasible to carry a package scheduled for an afternoon fight all day in the truck. In addition to those, there are many other operational constraints and regulations motivating the multi-trip vehicle routes. Companies may need to limit the total value of the items they carry on a vehicle due to insurance purposes. For example, one of the services of Brinks is providing cash distribution to ATM's from banks. In such kind of operations, lowering the risk of theft becomes as important as providing a fast service, which motivates carrying less on a vehicle and operating multiple trips. Similarly, safety regulations in city logistics may also require making the distributions in smaller amounts. For example, distribution of oil to gas stations is subject to such regulations.

3. A SET COVERING-BASED CONSTRUCTION MATHEURISTIC

We represent the problem on a graph $G=(N, A)$ where $N=\{0,\ldots, n\}$ is the set of nodes and A is the set or arcs. Node 0 corresponds to the depot while others correspond to customers.

The matheuristic we propose uses a set covering approach: We frst construct a set of feasible routes, J, and then choose a subset to ensure that each customer is visited once while obeying the allowable service time for each vehicle with minimum cost.

While generating the set of feasible routes, a route is allowed to have either only a delivery or pickup nodes to align with the company operations. Managing the heterogeneous vehicle capacities requires partitioning the route set J. The capacity-based partition is as follows: $J=\{1,\ldots, r^1, r^1+1, \ldots, r^2, r^2+1, \ldots, r^3\}$, where $J_p=\{1, \ldots, r^p\}$ is the set of routes that can be assigned to vehicles of type p (p^{th} smallest capacity). Note that $J_1 \subseteq J_2 \subseteq J_3$ since each route in J_1 that can be served by a vehicle having the smallest capacity can also be served by a vehicle with medium or large capacity as well. The set covering formulation ensures the optimal solution if J includes all feasible routes, yet solving the problem may not be feasible considering the computation-time requirements in practice. Therefore an effort must be made to choose small number of high-quality routes into set J. For a given set of feasible routes J, the set covering [SC] formulation for the problem described in the previous section is as follows:

$$[SC] \quad \min \sum_{k\in K} \alpha_k \qquad (1)$$

s. to

$$\sum_{j\in J}\sum_{k\in K}\sum_{h\in H} \alpha_{ij} x_{jk}^h \geq 1 \qquad \forall i \in N_1 \cup N_2 \qquad (2)$$

$$\sum_{j\in J} x_{jk}^h \leq 1 \qquad \forall h \in H, k \in K \qquad (3)$$

$$x_{jk}^h \geq x_{jk}^{h+1} \qquad \forall h \in H, k \in K, j \in J \qquad (4)$$

$$\tau_k^h + t_j - M(1 - x_{jk}^h) \leq \tau_k^{h+1} \qquad \forall h \in H, k \in K, j \in J \qquad (5)$$

$$e_j x_{jk}^h \leq \tau_k^h \leq l_j x_{jk}^h \qquad \forall h \in H, k \in K, j \in J \qquad (6)$$

$$\tau_k^h + t_j - M(1 - x_{jk}^h) \leq T + \alpha_k \qquad \forall h \in H, k \in K, j \in J \qquad (7)$$

$$x_{jk}^h = 0 \qquad \forall h \in H, k \in \{1, \dots, k^1\}, j \in J \setminus J_1 \qquad (8)$$

$$x_{jk}^h = 0 \qquad \forall h \in H, k \in \{1, \dots, k^2\}, j \in J \setminus J_2 \qquad (9)$$

$$\tau_k^h \geq 0 \qquad \forall h \in H, k \in \{1, \dots, k^2\} \qquad (10)$$

$$x_{jk}^h \in \{0, 1\} \qquad \forall h \in H, k \in K, j \in J \qquad (11)$$

$$\alpha_k \geq 0 \qquad \forall h \in H, k \in K, j \in J \qquad (12)$$

where parameter t_j is the duration of route j, e_j and l_j are the earliest and latest dispatching times of route j, T is the total allowable duration of routes assigned to a vehicle, K is the number of vehicles, H is the maximum number of routes that can be assigned to a vehicle. By slightly abusing the notation, we use K and H also for the set of vehicles and set of routes that can be assigned to a vehicle, respectively. Recall that N_1 is the set of nodes with positive demand and N_2 is the set of nodes with negative demand. a_{ij} is a binary parameter whose value is 1 if node i is on route j, 0 otherwise. M is a big number. x_{jk}^h is the binary decision variable that equals 1 if vehicle k serves on route j in the h^{th} order within its tour. The nonnegative continuous decision variables are τ_k^h and α_k. τ_k^h is the starting time of the h^{th} route served by vehicle k and α_k denotes the maximum allowable overtime for vehicle k. The objective function (1) minimizes the total overtime. Constraints (2) impose visiting each customer node. Constraints (3) assign at most one route to a position. Constraints (4) prevent assigning a route to position h +1 before any assignment to h is done. Constraints (5)-(7) ensure that each route starts between its earliest and latest dispatching time. Constraints (8) and (9) ensure the load-vehicle compatibility. Constraints (10) and (12) are nonnegativity and constraints (11) are the binary set constraints.

Note that in SC model, as suggested by the real routes provided by Santa Fe in the next section, we assume that the time windows are large enough such that changes in the truck dispatching time do not affect the duration of a route. Otherwise, depending on the truck dispatching time τ_k^h, waiting at a customer node would differ and constraints (5)-(7) would not work as intended, so necessary adjustments should be carried out.

4. COMPUTATIONAL RESULTS

In this section, we present the computational results of the matheuristic based on an SC approach. Since there is no study in the literature which focuses on the same SC problem faced by Santa Fe, we do not have a direct comparison basis. Therefore, we simplify our model ignoring the time windows and assuming a homogeneous vehicle feet. We compare our results with the best solutions presented in Rochat and Taillard [10] on some instances so that we can develop a sense of performance of the SC approach.

We frst describe the methods we use to create the set of feasible routes J, and then present the simplified SC formulation. The frst group of methods we use to generate feasible routes are the ones from literature. These are namely savings heuristics [5], insertion heuristic [12] and route-frst cluster-second method. In route-frst cluster-second method, given a VRPM instance, we frst solve the traveling salesman problem (TSP) in the route-frst phase for the entire node set. Then, following the order of the nodes on the TSP tour and obeying the vehicle capacity and maximum duration constraints, we create the feasible routes. That is, each feasible route r_{ij} which starts from the depot and visits all the nodes between $(i+1)^{\text{th}}$ and j^{th} nodes on the giant tour is accepted to the set J, where $0 \leq i < j$. In addition to these classical methods, we also use some new methods to enrich the feasible route set J avoiding to increase the cardinality undesirably. These methods can be listed as follows.

1. Modified savings algorithm: Instead of computing the savings just considering the distances as in the classical savings algorithm, we also consider the polar angles between the nodes and the depot while merging the routes with the aim of decreasing the zigzags around the depot. We compute the savings in this modified version as follows:

$$s_{ip} = \begin{cases} d_{0i} + d_{0p} - d_{ip}, & if\ \theta_i = \theta_p \\ \left(d_{0i} + d_{0p} - d_{ip}\right)\frac{2\Pi}{\theta_i - \theta_p}, & otherwise \end{cases} \quad (13)$$

where d_{ip} is the distance between nodes i and p and θ_i is the polar angle between node i and the depot.

2. Distance -polar angle insertion: While generating the routes using the insertion method, we frst initialize a route, then at each iteration we determine the next node u to insert between nodes i and p on the existing routes. In our insertion methods, we use different rules than the ones in the literature. Distance - polar angle insertion method considers the polar angle difference between the nodes i and u in addition to the increase in the distance caused by this insertion. The criterion we use for this insertion rule is as follows:

$$c_{ip}^{u} = (d_{iu} + d_{up} - d_{ip})(\theta_i - \theta_u) \quad (14)$$

At each iteration, we choose the nodes (i, u, p) with the minimum c_{ip}^{u} for insertion.

3. Segment -polar angle insertion: In this method, we define a new criterion to determine the next node to insert into the existing routes. Assuming that all the nodes in the node set are enclosed with a circle whose diameter is the maximum node-to-node distance, we try to minimize the movements between the segments along with the movements around the polar angle with the depot. For this purpose, we use the following criterion to measure the movement while inserting node u between nodes and is as follows:

$$c_{ij}^{u} = \frac{|d_{0i} - d_{0u}||\theta_i - \theta_u| + |d_{0j} - d_{0u}||\theta_j - \theta_u| - |d_{0i} - d_{0j}||\theta_i - \theta_j|}{d_{max}\theta_{max}} \quad (15)$$

where d_{max} is the maximum distance in the graph and θ_{max} is the maximum polar angle between the customer nodes and the depot node.

4. Route -patching: In the SC formulation, we try to keep the number of routes in set as small as possible due to computation-time concerns. Even though small number of arcs can result in infeasibility, increasing the number of routes to ensure feasible solutions can be avoided by generating *node-patches* for the existing routes. We create the node-patches as follows. After generating the routes according to the aforementioned rules, for each node u we search for a possible insertion with the minimum-cost. The cost of insertion is the increase in the distance. Node u can be patched to a current route only if the vehicle capacity and the allowable route duration limit are not violated in the case of patching and it is not already on that route. By using node-patching we avoid generating more routes and storing more data. We denote the set of node-patches by J^P.

Having defined the route generation rules, the simplified SC formulation is as follows:

$$\text{[Simplified SC]} \quad \min \sum_{j \in J} \sum_{k \in K} c_j x_j^k \quad (16)$$

s. to

$$\sum_{j \in J} \sum_{k \in K} \alpha_{ij} x_j^k \geq 1 \qquad \forall i \in V \quad (17)$$

$$\sum_{j \in J} c_j x_j^k \leq T \qquad \forall k \in K \quad (18)$$

$$x_r^k \leq x_j^k + (1 - b_{rj}) \qquad \forall r \in J^P, k \in K, j \in J \quad (19)$$

$$x_j^k w_j + \sum_{r \in J^P} \left(w_r b_{rj} x_r^k\right) \leq C \qquad \forall r \in J^P, k \in K, j \in J \quad (20)$$

$$x_j^k \in \{0, 1\} \qquad \forall\, k \in K, j \in J \quad (21)$$

where b_{rj} is a binary parameter which equals to 1 if route r is patched to route j, 0 otherwise, $c_j(w_j)$ is the total cost (demand) of the customers on route j, C is the vehicle capacity, V is the set of nodes. x_j^k is the binary decision variable that equals to 1 if route j is assigned to vehicle k. The objective function (16) minimizes the travel distance. Constraints (17) make sure that each node is visited, and constraints (18) ensure that allowable duration for a vehicle is obeyed. Constraints (19) impose that a patched route cannot be chosen unless the route it is patched to is chosen. Constraints (20) satisfy that the total demand on a route and its patches will not exceed the vehicle capacity. Constraints (21) are binary set constraints.

We compared the construction results of the simplified version of our set covering formulation with Rochat and Taillard [10] on the nine instances from the literature shown in Table 1. Algorithms are coded in C and the problems are solved using Cplex 12.6 on Xeon X5660 or near equivalent processors.

To generate the routes for the simplified SC model, we run the six aforementioned heuristics once for capacity C. Since the construction heuristics try to insert a node into an existing route as long as the constraints are not violated, the final routes' loads are close to the maximum allowable capacity. Therefore, it is difficult to patch a node to these routes. To create more opportunities for node-patching, we created additional routes using smaller capacities. We tested different combinations of different capacities for the six construction heuristics. To generate additional routes, we ran the savings heuristic with $0.8C$ and Solomon's insertion heuristic [12] with $0.5C$ and $0.8C$ as we obtained the best results on average among the others. We also used the entire set of routes generated by the set partitioning method. The number of routes generated is determined by the number of nodes, capacity of vehicles, demand of each node, distances between the nodes, maximum allowable driving duration and number of node patched. The number of routes for each run is also provided in the Table 2 below for $K = 1$. z^{SC}. denotes the objective function value of the model

Table 1. Test Instances From the Literature

Instance	Number of customers	Vehicle Capacity
CMT1	50	160
CMT2	75	140
CMT3	100	200
CMT4	150	200
CMT5	199	200
CMT11	120	200
CMT12	100	200
F11	71	30,000
F12	134	2,210

based on the SC approach, $t(sec)$ is the computation time in seconds, and #routes is the cardinality of the route set J.

In these experiments, as the allowable driving duration T, we used the results found by Rochat and Taillard with 10% slack, i.e., $T = 1.1r^{RT}$, where z^{RT} is the solution obtained by Rochat and Taillard [10]. Despite being limited to a small number of routes, simplified SC model can still find good initial solutions in seconds. Moreover, we see that creating node-patches can improve the solution. Comparing to Rochat and Taillard [10], we see that the SC model can create good initial solutions, which would make the improvement process faster especially on larger instances.

Table 3 provides further computational results for different numbers of vehicles. We can still find solutions in less than a second for many instances.

Even though our focus is on a more involved problem, the computational results of the simplified model show that the SC approach can provide good initial solutions within very short times. This stands out as an advantage of the SC approach since the problem described in Santa Fe case study is a computationally more demanding version of the VRPM.

Having shown that the SC-based matheuristic is promising for the VRPM, we now provide computational results for Santa Fe instances, which are small in size, and the larger instances we obtained by modifying the instances in the literature. An example data for a typical working day of Santa Fe is given in Table 4. The type column specifies which kind of service is requested by the customer. *Load* is the space requested. *Time windows* indicate when the customer is available. Driving time is how long it takes from the depot to the customer location. *Service time* is the estimated material handling time at the customer site.

In Table 5, we provide the computational results. In this table, N is the number of customers, #routes is the cardinality of route set, which is created by accepting all feasible routes due to small number of customers, U% is the fleet utilization rate, as the ratio of the total operational time to the total time available, and t (sec) as the computation-time to solve the problem using CPLEX 10.

Table 2. Comparison of Simplified SC Model with Rochat and Taillard [10] with Different Number of Patches

Instance	z^{RT}[10]	Simplified SC (1patch)			Simplified SC (7 patches)		
		z^{SC}	$t(sec)$	#routes	z^{SC}	$t(sec)$	#routes
CMT1	524.61	579.11*	2.99	557	571.59	3.32	857
CMT2	835.26	907.39	0.06	716	903.26	4.20	1166
CMT3	826.14	886.83	0.17	1425	886.83	0.90	2025
CMT4	1028.42	1134.74*	0.21	2156	1134.74*	5.22	3056
CMT5	1291.44	1395.74	0.26	2717	1395.74	4.21	3911
CMT11	1042.11	1071.07	0.17	2069	1068.09	0.26	2789
CMT12	819.56	828.59	0.07	1202	825.87	0.59	1802
F11	241.97	256.19	0.13	1295	256.19	0.31	1721
F12	1162.96	1219.32	0.32	2916	1219.32	8.07	3765

(* $T = 1.15z^{RT}/K$)

Table 3. Computational Results of Simplified SC Model for Different Number of Vehicles for one Node-Patch

Instance	K	T	z^{SC}	$t(sec)$	#routes
	1	604	579.11*	0.24	557
	2	303	579.11*	0.99	557
CMT1	3	202	597.38*	5.59	559
	1	919	907.39	0.06	716
	2	460	907.39	3.51	716
CMT2	3	307	907.39	10.26	716
	1	909	886.83	0.17	1425
	2	455	886.83	0.59	1425
CMT3	3	303	886.83	1.11	1425
	1	1198	1134.74	0.4	2156
	2	599	1134.74	2.19	2156
CMT4	3	400	1134.74	3.31	2156
	1	1421	1395.74	0.26	2717
	2	710	1395.74	0.86	2717
CMT5	3	-	-	-	-
	1	1147	1071.07	0.17	2069
	2	573	1071.07	0.76	2069
CMT11	3	382	1071.07	23.11	2070
	1	902	828.59	0.07	1202
	2	451	828.59	0.32	1202
CMT12	3	301	828.59	0.68	1202
	1	267	256.19	0.13	1295
	2	133	256.19	0.43	1297
F11	3	89	256.19	0.57	1141
	1	1280	1219.32	0.32	2961
	2	640	1219.32	1.47	2961
F12	3	-	-	-	-

(* $T = 1.15z^{RT}/K$)

Table 4. A Typical Working Day of Santa Fe

Customer	Type	Load (liftvan)	Time window	Driving time (min)	Service time (min)
1	Pick-up	4	9am - 3pm	30	90
2	Delivery	3	8am - 11am	60	60
3	Pick-up	3	12am - 4pm	60	75
4	Pick-up	1	8am - 11am	30	60
5	Delivery	1	12am - 4pm	60	90
6	Pick-up	2	8am - 4pm	90	30
7	Pick-up	2	8am - 4pm	90	30
8	Pick-up	2	8am - 4pm	90	30
9	Delivery	3	8am - 4pm	30	30
10	Delivery	3	8am - 4pm	30	30
11	Delivery	1	10am - 6pm	90	60

Table 5. Results of Scheduling Santa Fe activities

Day	N	#routes	$U(\%)$	$t(sec)$
1	11	22	66	6.2
2	7	18	38	1.2
3	8	20	51	1.6
4	7	18	35	1

Since the Santa Fe instances are small in size, we performed further computational experiments to better observe the behavior of our SC formulation on larger-sized problems with similar attributes. For these experiments, we generated new instances by modifying CMT1 and CMT11. The reason we chose these two instances to modify is that CMT1 is representative of randomly distributed nodes and CMT11 is representative of clustered nodes (Figures 1 and 2).

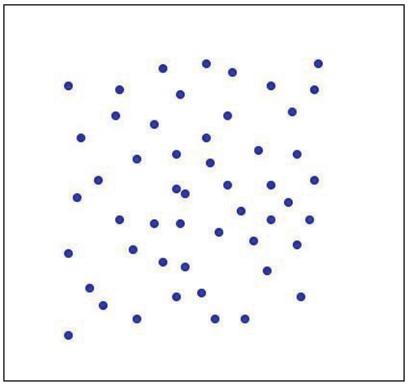

Figure 1. CMT11

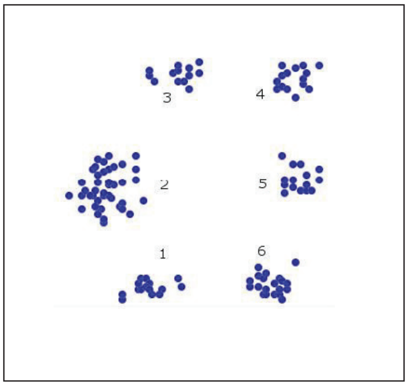

Figure 2. CMT1

For each instance, we divided the nodes into two groups such that one group contains the nodes for delivery and the other group contains the nodes for pickup. This division is done in three different ways and we name the new instances as $CMT1_1$, $CMT1_2$, $CMT1_3$, $CMT11_1$, $CMT11_2$, and $CMT11_3$.

To create $CMT1_1$, $CMT1_2$ and $CMT1_3$, the first 15, 25 and 35 customers of CMT1 are assumed to request delivery while the rest are assumed to request a pickup service respectively. To create the variants of CMT11, customers in the clusters 1-2-3, 1-3-5 and 2-4-6 are assumed to request a delivery while the rest request a pickup service. For these new instances, we assume that there are two vehicles available with capacities $C[1]$ and $C[2]$, where $C[1] + C[2]$ is equal to the original vehicle capacity for the corresponding instance, and we use the same T as before. We do not include the time window attribute since node patches cannot be used when there are time windows. The results are shown in Tables 6 and 7. For these instances, the optimal solution values are not know. Our observations regarding the behavior of the SC-based model are as follows. Required computation

Table 6. SC Results on Versions of CMT1

Instance	Patch	$C[1]$	$C[2]$	z^{SC}	$t^{SC}(sec)$	#routes
	1	50	110	812.35	1.02	514
	5	50	110	812.35	3.67	714
$CMT1_1$	15	50	110	812.35	26.07	1214
	1	60	100	840.26	1.73	475
	5	60	100	839.32	3.6	1075
$CMT1_1$	15	60	100	828.73	37.95	1175
	1	70	90	842.74	0.93	434
	5	70	90	842.74	4.62	634
$CMT1_1$	15	70	90	839.94	99.29	1134
	1	50	110	837.64	1.57	494
	5	50	110	837.64	5.51	694
$CMT1_2$	15	50	110	815.64	21.22	1194
	1	60	100	831.75	1.9	459
	5	60	100	831.75	3.84	659
$CMT1_2$	15	60	100	821.75	35.66	1159
	1	70	90	841.41	0.67	420
	5	70	90	839.62	4.48	620
$CMT1_2$	15	70	90	837.74	65.21	1120
	1	50	110	861.87	1.31	492
	5	50	110	859.19	4.73	692
$CMT1_3$	15	50	110	857.02	50.84	942
	1	60	100	871.25	1.6	456
	5	60	100	868.28	3.76	656
$CMT1_3$	15	60	100	857.4	43.7	1156
	1	70	90	889.26	1.21	421
	5	70	90	884.44	4.5	621
$CMT1_3$	15	70	90	876.29	80.39	1121

times to solve the SC model on these instances are still very short. Different capacity allocations result in different solutions; yet, we do not observe a pattern between the change in the vehicle capacities and the solution values. Using node-patches helps improving the solution for randomly distributed nodes (CMT1), while it has very little effect in the clustered case (CMT11). The reason is that when the nodes are clustered, the routes are more likely to be composed of the nodes belonging to the same cluster. In that case, the patching procedure will try to patch a node from another cluster and fail due to exceeding the travel duration limit. Therefore, it becomes more difficult to find a proper patch when the nodes are clustered.

5. CONCLUSION

Motivated by a real-life application provided by a distribution company in Jakarta and lack of a solution method in the literature for involved multi-trip vehicle routing problems with different attributes, in this paper we study a variant of the VRPM. We present a cons-

Table 7. SC Results on Versions of CMT11

Instance	Patch	$C[1]$	$C[2]$	z^{SC}	$t^{SC}(sec)$	#routes
	1	50	150	1624.01	25.45	1785
	5	50	150	1624.01	39.5	2265
$CMT11_1$	10	50	150	1623.52	68.64	2865
	1	70	130	1693.84	17.89	1546
	5	70	130	1693.84	33.77	2026
$CMT11_1$	10	70	130	1693.45	64.98	2626
	1	90	110	1805.3	13.48	1346
	5	90	110	1805.3	27.65	1826
$CMT11_1$	10	90	110	1805.17	55.3	2426
	1	50	150	1729.18	25.48	1780
	5	50	150	1728.92	43.91	2260
$CMT11_2$	10	50	150	1728.92	75.59	2860
	1	70	130	1726.64	18.14	1558
	5	70	130	1726.64	34.21	2038
$CMT11_2$	10	70	130	1726.64	80.94	2638
	1	90	110	1807.02	14.56	1341
	5	90	110	1802.41	27.89	1821
$CMT11_2$	10	90	110	1798.13	59.5	2421
	1	50	150	1605.53	24.55	1800
	5	50	150	1602.4	39.36	2280
$CMT11_3$	10	50	150	1601.44	65.65	2880
	1	70	130	1674.31	18.71	1571
	5	70	130	1674.31	34.49	2051
$CMT11_3$	10	70	130	1671.33	60.97	2651
	1	90	110	1817.17	14.4	1370
	5	90	110	1816.87	29.12	1850
$CMT11_3$	10	90	110	1811.55	163.43	2450

truction matheuristic based on set covering approach. Since this is the frst study on the VRPM with time windows and delivery and pickup, we did not have a direct comparison basis in the literature. Therefore, we simplified our approach and compared it with Rochat and Taillard's [10]. This comparison showed that our SC approach can provide good solutions in seconds. We presented the results for the real-life instances provided by Santa Fe and also on the instances which are created by modifying the instances from the literature which are larger in size. Even though this is not the frst time a set covering approach is considered as a solution method in a routing problem, to the best of our knowledge this is the frst study integrating different attributes in a VRPM

REFERENCES

1. **Azi, N., Gendreau, M., Potvin J. Y.** 2010. "An Exact Algorithm for a Vehicle Routing Problem with Time Windows and Multiple use of Vehicles", European Journal of Operational Research, 202 (3): 756-763.
2. **Azi, N., Gendreau, M., Potvin J. Y.** 2014. "An Adaptive Large Neighborhood Search for a Vehicle Routing Problem with Multiple Routes", Computers & Operations Research, 41: 167-173.
3. **Brandao, J.**, **Mercer, A.** 1997. "A Tabu Search Algorithm for the Multi-Trip Vehicle Routing and Scheduling Problem", European Journal of Operational Research, 100 (1): 180-191.
4. **Cattaruzza, D., Absi, N., Feillet, D., Vidal, T.** 2014. "A Memetic Algorithm for the Multi Trip Vehicle Routing Problem", European Journal of Operational Research, 236 (3): 833-848.
5. **Clarke, G. U.**, **Wright, J. W.** 1964. "Scheduling of Vehicles From a Central Depot to a Number of Delivery points", Operations Research, 12 (4): 568-581.
6. **Hernandez, F., Feillet, D., Giroudeau, R., Naud, O.,** 2014. "A New Exact Algorithm to Solve the Multi-trip Vehicle Routing Problem with Time Windows and Limited Duration", 4OR, 12 (3): 235-259.
7. **Mingozzi, A., Roberti, R., Toth, P.** 2013. "An Exact Algorithm for the Multitrip Vehicle Routing Problem", INFORMS Journal on Computing, 25 (2): 193-207.
8. **Olivera, A.**, **Viera, O.** 2007. "Adaptive Memory Programming for the Vehicle Routing Problem with Multiple Trips", Computers & Operations Research, 34 (1): 28-47.
9. **Prins, C.** 2004. "A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem", Computers & Operations Research, 31 (12): 1985-2002.
10. **Rochat, Y., Taillard, E. D.** 1995. "Probabilistic Ddiversification and Intensification in Local Search for Vehicle Routing", Journal of Heuristics, 1 (1): 147-167.
11. **Salhi, S., Petch, R. J.** 2007. "A GA Based Heuristic for the Vehicle Routing Problem with Multiple Trips", Journal of Mathematical Modelling and Algorithms, 6 (4): 591-613.
12. **Solomon, M. M.** 1987. "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", Operations Research, 35 (2): 254-265.
13. **Taillard, E. D., Laporte, G., Gendreau, M.** 1996. "Vehicle Routing with Multiple use of Vehicles", Journal of the Operational Research Society, 47 (8): 1065-1070.
14. **Yang, Y., Tang, L.** 2010. "A Filter-and-fan Approach to the Multi-Trip Vehicle Routing Problem", In Logistics Systems and Intelligent Management, 2010 International Conference, IEEE.