MJFAS Malaysian Journal of Fundamental and Applied Sciences
ISSN:2289-5981 e-ISSN:2289-599X
UTM UNIVERSITI TEKNOLOGI MALAYSIA

Vehicle Routing Problem with Time Constraints

Farhana Johar[a], Chris Potts[b], Julia Bennell[c] *

[a] *Department of Mathematical Sciences, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia.*
[b] *School of Mathematics, Faculty of Social and Human Sciences, University of Southampton, SO17 1BJ United Kingdom.*
[c] *School of Mathematics, Faculty of Social and Human Sciences, University of Southampton, SO17 1BJ United Kingdom.*

**Corresponding Author: farhanajohar@utm.my*

Article history :
Received 26 November 2015
Accepted 22 December 2015

GRAPHICAL ABSTRACT

ABSTRACT

This paper addresses the Vehicle Routing Problem (VRP) with time constraints which been solved by several heuristic algorithms. The problem starting at the depot where the customer orders which associated with due date determined by customer, are released with different point of time. Ideally, to avoid any lateness in delivery process, the orders need to be delivered as soon as it released and available at the depot. However, this may increase the traveling cost because one vehicle needs to go and come back to depot for the other deliveries which this can be saved by batching the deliveries. Therefore, the study will focus on minimizing the tradeoff between traveling and tardiness costs. Literatures show that implementing the heuristic algorithms for solving various instances of VRPs manage to minimize the distribution cost within the reasonable computing times. An initial feasible solution was generated using a constructive heuristic. The solution then was improved by several metaheuristic algorithms were developed for solving the problem studied; Variable Neighborhood Search, Large Neighborhood Search and Tabu Search. To cater with the problem studied, a modification to the benchmark problems of Solomon has been done. The performance of the algorithms can be seen through the comparison of the solution obtained. The results showed that there is a significant saving in producing the least cost solution and manually constructed routes are very ineffective.

Keywords: vehicle routing problem, metaheuristics, local search

http://dx.doi.org/10.11113/mjfas.v11n4.400

1. INTRODUCTION

Vehicle routing problem (VRP) has become more crucial in real world application nowadays. Consumers are really known what the best for them. Suppliers also needs to know how to cater the customer's need in optimal way, so the customers can get the best out of it and so do suppliers. The good vehicle routing will lead to a win-win situation between suppliers and the customers. Latest survey on the VRP and its variants was returned by Kumar and Panneerselvam [1].

The vehicle routing problem with time windows (VRPTW) is one of the important variants in VRP which encountered very frequently in making decisions about the distribution of goods and services. In VRPTW, a set of vehicles with limited capacity need to be routed from a central depot to a set of geographically dispersed customers with known demand within customer's predefined time window. This time window is the interval time for customer to be served. Kim and Sahoo [2], Balakrishnan [3], Chen *et al.* [4] and Min [5] classified time window into two; soft and hard time window. In soft time window (VRPSTW), a vehicle can break customer's time window with penalty cost included. However, in VRP with hard time window have to be followed strictly [4].

Due to its importance in real life, the VRPTW continues to draw attention from researchers and has become a well-known problem in VRP literature. For an overview, readers are referred to the surveys conducted by Bodin *et al.* [6] in which provided a comprehensive review of the original VRPs, i.e. the problem without time windows, and assessed the published work that related to algorithmic developments; Solomon [7], Cordeau *et al.* [8] and, Braysy and Gendreau [9,10]. Among these, Solomon was the first to generalize VRP heuristics for solving VRPTW [11].

Recent work on VRPTW can be found in Figliozzi [12] which presented an algorithm that can tackle time dependent VRP with hard and soft time windows without alteration in its structure. Review of some limited of exacts, heuristics and metaheuristics methods for VRPTW are given in El-Sherbeny [13]. Taner *et al.*[14] also used metaheuristic algorithm as their approach to solve the VRPTW same as Minocha and Tripathi [15] which proposed a two phase method which utilizes genetic algorithms as well as random search incorporating simulated annealing concepts.

Since we will use variable neighborhood search (VNS), large neighborhood search (LNS) and tabu search (TS) to solve our vehicle routing problem, we have done some reviews on the three metaheuristics algorithms which can be used to solve different variants of VRP. VNS is a recent metaheuristic which exploits systematically the idea of neighbourhood change [16] in the decent to local minima and in the escape from the valleys which contain them. It was an algorithm that combines local search strategy and the variable neighbourhood structure [17]. Stenger *et al.* [18] presented a new adaptive VNS (AVNS) for multi-depot VRP with private fleet and common carriers (MDVRPPC).

LNS improves the current solutions by using its searching process in large neighborhoods, which may contain potentially better solutions. The algorithm works by removing a large number of customers from their current routes and re-insert them into other routes at optimal cost. We provide the following literatures to show that LNS was used in different variant of VRPTW. Hong [18] proposed an improved LNS algorithm to solve the dynamic VRPHTW (DVRPHTW). Korsvik et al. [19] also proposed same algorithm to solve the tramp ship routing and scheduling problem with split loads. A computational study was performed and it shows that introducing split loads can yield significant improvements. Furthermore Kovacs et al. [20] developed a template based adaptive large neighborhood search (TALNS) which integrates several problem-specific destruction and repair sub-heuristics to solve the consistent VRP (ConVRP).

Another metaheuristic approaches that we will used to solve our VRP is TS. TS is a memory-based local search metaheuristic. It was proposed by Fred Glover in 1986 [21] to improve local search by accepting non-improving moves within the neighbourhood. TS is claimed as among of the most effective method and somehow the best to tackle the difficult problems at hand [22], [23]. TS also can be used to solve different variants of VRPTW. Moccia et al. [24] described a new incremental tabu search (ITS) heuristic that built upon a previously developed TS by replacing its neighbourhood structure for the generalized vehicle routing problem with time windows (GVRPTW). To solve VRPTW with simultaneous delivery and pickup in home health care, Liu et al. [25] proposed a Genetic Algorithm (GA) and TS based on route assignment attributes of patients, an augmented cost function, route re-optimization, and attribute based aspiration levels were proposed in the paper. Ceschia et al. [26] also proposed an approach based on TS and on a combination of neighborhood relations for solving another variant of VRPTW, heterogeneous vehicle routing problem with time windows and carrier-dependent costs.

In real world application, transport operators always face with the limited resources problem such as fixed fleet, *m*. To our best knowledge, there has been little research on limited vehicle routing problem and the earliest related research can be found in Lau et al. [27]. The next research on limited vehicles was carried out by Lim and Wang [28]. The objective is to maximize the number of served customer in the route. In solving the same problem, Lim and Zhang [29] presented a two-stage algorithm of *m*-VRPTW which focuses on maximizing the number of served customer at the first stage and an iterated multi-start hill climbing algorithm with classical and newly defined operators including Generalized Ejection Chains (GEC) was used to minimize the total distance travelled. Latest work on *m*-VRPTW was presented by Wang et al. [30]. The authors designed the customer clustering assignment algorithm and genetic algorithm to maximize the number of served customers and minimizing the distance travelled.

Our VRP has dealt with release date and due dates. To date, literature on VRP with due date (VRPDD) is relatively small. Park [31] studied vehicle scheduling problem with due dates and time deadlines. Penalty is given when time taken to visit a customer exceeds the customer's due times. However, this must be within their time deadline. The objectives are to minimize the total travel time, total weighted tardiness and fleet size. Other study on VRPDD was carried by Kang et al. [32] with the objective to minimize the weighted sum of the traveling time and total tardiness.

2. TIME DEPENDENT VEHICLE ROUTING PROBLEM (TDVRP)

We studied one of the VRPs which can be categorized as one of variations in VRPSTW. Our VRP studied; time dependent vehicle routing problem (TDVRP) is depending on the time of order released from the production line. The vehicles will only depart from the depot as soon as all order's scheduled in the vehicle have been released from the line. Therefore, sequence in the production line will determine the release date of each customer's order. We make use the output from production line; i.e. release date as time constraints in our VRP.

The study on this topic is motivated by the make-to-order business trend production-distribution coordination problem encountered in the supply chain where the most problem faced by the company is to find a joint schedule of production and distribution by taking into account both customer service level and total distribution costs.

In the following we describe the model of the problem studied. There are one supplier/manufacturer and N customers, $v_1, v_2, ..., v_N$ located at different locations in a given production-distribution system. At the beginning of a planning horizon, the supplier receives N orders/jobs, $q_1, q_2, ..., q_N$ requesting for processing in which each customer is associated with an order with the known size. Each order, q_i has a processing time p_i and latest time to be delivered at customer's location, due dates d_i.

All orders are to be processed on a single production line at the manufacturer/depot. It is assumed that one job is processed at a time and one after another. No preemption is allowed. We are not looking at the production scheduling stage in detail such as the setup cost or setup time for machine scheduling and so forth. At the production part, we concern about the order of jobs production in which release dates are generated.

Completed orders will deliver shortly to the customers by K available vehicles, which have same capacity limit, Q. The vehicle capacity limit is referred as the maximum total size of the jobs that can be delivered. It is also assumed that each vehicle is only using one route where it starts and ends at the depot, v_0. Each customer is only being served once and only by one vehicle.

Although released order is immediately loading into vehicle, one job might have to wait for others that been scheduled in the same vehicle due to varies of release date. Therefore, the vehicles can only leave the depot after all orders scheduled on the route are available. Hence, there are possibilities of having jobs late which a penalty is incurred. The penalty is given as a cost which can represent the cost of lost sales, or goodwill due to the customer inconvenience for not meeting the customer's due date. In this study, a linear loss function will be used, i.e. a penalty per unit time of tardy delivery.

3. PROBLEM FORMULATION

Our vehicle routing problem can be written in a graph theoretical perspective. Let $G = (V, A)$ be a complete undirected graph with vertex set is given as $V = \{v_0, v_1, ..., v_N\}$ and an edge set $A = \{v_i, v_j : v_i, v_j \in V, i > j\}$. Each vertex $v_i \in V$ is associated with:

- a known order size to be delivered, q_i ($q_0 = 0$ for depot)
- a processing time of each customer's order, p_i
- a release date of each customer's order, r_i
- a due date for each customer, d_i, where d_i is the latest time for customer to be served. Any servicing beyond the date will be considered as lateness in delivery.

Decision Variables:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ travels from customer } i \text{ to customer } j \\ 0 & \text{otherwise} \end{cases}$$

y_{ik} tardiness when vehicle k arrives at customer i, $y_{ik} \geq 0$

z_{ik} arrival time of vehicle k at customer i, $z_{ik} \geq 0$

z_{0k} departure time of vehicle k from depot, $z_{0k} \geq 0$

Parameters:

N maximum number of customers

K maximum number of vehicles

Q capacity of vehicle

q_i demand of customer i

s_i service time of customer i where $s_0 = 0$

d_i due time of customer i

r_i release time of demand of customer i

t_{ij} travel time from customer i to customer j

c_{ij} travel cost from customer i to customer j

w_i tardiness penalty cost of customer i

M large positive integer number

Formulation:

$$\min\ (\chi)\left(\sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=1}^{K} c_{ij}\, x_{ijk}\right) + (1-\chi)\left(\sum_{i=0}^{N}\sum_{j=0}^{N} w_i\, y_{ik}\right) \qquad (1)$$

subject to:

$$\sum_{j=1}^{N} x_{0jk} \leq 1 \quad \forall k \in K \qquad (2)$$

$$\sum_{i=1}^{N} x_{i0k} \leq 1 \quad \forall k \in K \qquad (3)$$

$$\sum_{k=1}^{K}\sum_{j=0, j\neq i}^{N} x_{ijk} = 1 \quad \forall i \in N \qquad (4)$$

$$\sum_{k=1}^{K}\sum_{i=0, i\neq j}^{N} x_{ijk} = 1 \quad \forall j \in N \qquad (5)$$

$$\sum_{i=1}^{N}\sum_{j=0, j\neq i}^{N} q_i x_{ijk} \leq Q \quad \forall k \in K \qquad (6)$$

$$z_{0k} \geq r_i \sum_{j=0}^{N} x_{ijk} \quad \forall i \in N, \forall k \in K \qquad (7)$$

$$z_{ik} + s_i + t_{ij} + M\left(x_{ijk} - 1\right) \leq z_{jk} \quad \forall i, j \in N, \forall k \in K \qquad (8)$$

$$z_{jk} + s_j - d_j + M\left(\sum_{i=0}^{N} x_{ijk} - 1\right) \leq y_{jk} \quad \forall j \in N, \forall k \in K \qquad (9)$$

4. HEURISTICS FOR TDVRP

Our VNS use the classical algorithm where random searching diversifies the search space in shaking phase. In VNS, a shaking phase used to escape the initial local optimum through the use of different neighborhoods. The shaking solution is then improved with a collection of neighborhoods in variable neighborhood decent.

LNS also used the random searching for relaxing a set of customers. Unlike in VNS, LNS will re-schedule a set of customers. The number of re-scheduled customers is increased one at a time if no further improvement has been found. Otherwise, the number will be reset to 1 and the process is continues until the maximum number is reached.

In contrast, the implementation of TS takes a systematic search in which is more efficient than ‘random walk’. TS is a good compromise of performance but always take a longer computational time. However, in this study, we use the same stopping condition as VNS to compare the performance of those heuristics.

Variable Neighborhood Search (VNS)

VNS is a metaheuristic for solving optimization problems based on systematic changes of structure within a search. The basic VNS consist of both diversification and intensification stages. Diversification is accomplished by applying shaking, i.e. random moves in larger neighborhoods, while local search component is included for intensification.

In the proposed VNS, relocate operator is used to shake the solution. Instead of relocate a customer, we are relocating several customers, say *k* simultaneously. To diversify the search, these *k* customers are selected randomly. The customers are removed from their current route one by one. The cost for each route is updated each time the customer is removed from the route. Then, re-insertion of removed customers into routes which selected randomly is done by using the least-cost insertion method. In this VNS, inserting a customer into their previous route is not allowed. We will first searching other possible routes to insert the customer without violating the vehicle capacity limit. A new empty vehicle is added if and only if no possible insertion can be done to any other existing vehicles due to the random insertion.

A solution obtained through shaking phase is submitted to local search procedure to search for the local optimal solution. In our local search, several neigborhoods were used to improve the shaking solution. There were relocate with different versions, swap, 2-opt and 3-opt. After going through the shaking and the local search improvement procedures have been performed, the obtained solution is compared to the incumbent solution. The comparison is done to determine whether the solution should be moved. A solution with an improved solution cost is always accepted.

Large Neighborhood Search (LNS)

LNS is based on a process of relaxation and re-optimization. The optimization method used is a simple greedy local search procedure. A set of customer visits are chosen and removed from the schedule and re-schedule at optimal cost. The process of going back and forth characterizes the move. The moves made within the neighborhood are generated in a randomized way. A move is accepted if it improves the cost solution, and rejected otherwise.

Our LNS will select the customers to be removed at random and rebuild the solution by inserting these customers into routes one at a time by using a greedy heuristics, i.e. least-cost insertion method. The insertion with the lowest cost will be chosen and the process will be repeating until all customers have been inserted.

Tabu Search (TS)

Tabu search is a metaheuristic that aims to proceed from a local optimum by allowing non-improving moves. A memory structure in the form of a tabu list prevents the same sequence of solutions being revisited. Our tabu list structure is created by make use of first-in first-out (FIFO) queue structure to store the tabu moves. This simple structure is as follows; <*TabuCust*, *TabuRoute*>.

This tells us that any move that shifting the customer *TabuCust* into route *TabuRoute* is forbidden for several times depending on the tabu list size, i.e. *TabuListMax*. Our Tabu Search's neighborhoods are based on some common move operators; relocate and exchange/swap. The best move from these two neighborhoods were improved by 2-Opt.

In each iteration, two different moves operators are applied to the current solution. From both on these neighborhoods, only the best non-tabu move is chosen. If the number of tabu move in the tabu list is less than *TabuListMax*, the move then is set tabu and will not be considered as valid move until the tabu status is expired.In the tabu search, all kind of solution is accepted even if it worst than before but the best solution so far is kept until the end.

5. DATA SETS AND PROBLEM DESIGNED

56 VRPTW 100-customer instances of Solomon benchmark problem have been modified and used throughout this study. We introduced three different problem types, which based on three different numbers of weights, χ . We are varying the values of χ as 0.9, 0.1 and 0.5, which designed the problem to focus on minimizing the traveling cost, tardiness cost and the total costs respectively.

Due to the fact that different orders may have different due dates, delivering more orders on time might require a large number of vehicles. Even though this will leads to higher distribution cost; i.e. traveling cost, at some point it might reduce the total tardiness time. This will gives some insight to find the best joint schedule of production and distribution.

This can be seen as one of the company's strategies in which the usage of resources to the max might improve the overall costs. Therefore, we are observing the impact of limited vehicles on our problem studied. To investigate on how much the company could get the benefit by maximizing the usage of their resources, the different fleet sizes have been designed by using the formula given in [33]. For each type of the fleet sizes, we based on the percentage of utilization of vehicle capacity, a.

We use three values for parameter a to designed three types of fleet sizes: $a = 0.3$, 0.5 and 0.7 which referred as big, medium and small–fleet size of vehicles respectively.

The TDVRP is variation of VRPTW. Therefore, due to the lack of real and benchmark cases for the TDVRP, our computational study uses modified benchmark data originating from the VRPTW which were taken from the 100-customer test instances of Solomon [7]. This dataset was used for all three of fleet sizes in which three different weights are applied.

6. COMPUTATIONAL RESULTS

We have experiment a few parameters to set our algorithms. From the VNS results, we found that it is the best to set *k=1* which been used in our shaking phase. Two setting on local search order have been tested which concluded that local search neighborhoods performed well in four neighborhoods in the following order; 3-Opt, 2-Opt, Swap and Relocate. In LNS, *K*=50 was determined as the best maximum number of re-scheduled customers. In tabu search, we have been looking for the appropriate tabu list size; i.e. *TabuListMax,* which agreed by all fleet size cases. The results concluded that *TabuListMax=20* is sufficient to prevent the searching process from cycling which might trapped in local optima.

Three methods were implemented with different parameters in order to obtain the best possible results. In VNS and LNS, the algorithms make use of random numbers. Hence, average results were obtained after making 10 runs for each instance in each weight and fleet size. All methods used here are coded in C++ and run on a desktop computer running on an Intel PentiumCore, 2.66Ghz processor with 16.0GB of RAM.

The benchmark VRPTW of 56s Solomon (1987) instances with some modification had been used as a platform for testing the algorithms. The problems are categorized into six classes, namely C1, C2, R1, R2, RC1 and RC2. Problems set with C categories are clustered data, meaning that locations of customer are geographically close to each other's. Problem with R categories are uniformly distributed data, and those from RC classes are hybrid problems that have features of both C and R categories. In addition, C1, R1, and RC1 problems sets have narrower time between release date and due date, whereas the other problem sets have wider time between the dates.

To have a fairer comparison, the iterations will be terminated after 30s of CPU times and the current best solution obtained is the solution to the method. It is understood that each method is developed on varying nature, so the duration of the test should be vary. In mean while, having the best solution cost among others but taken much longer time is not the indicator to conclude certain particular method is the best. Therefore we use the same stopping condition in all methods used.

The solution found in each algorithm has been shown in the following Table 1.

Table 1 Computational Results of All Problem Designed.

Big Fleet Size, a=0.3									
Class	Focus on Tardiness, w=0.1			Focus on both costs, w=0.5			Focus on Traveling, w=0.9		
	VNS	LNS	TS	VNS	LNS	TS	VNS	LNS	TS
C1	581.09	587.26	572.87	1493.05	1484.63	1436.04	1846.23	1846.40	1897.75
C2	107.63	108.19	109.98	539.40	539.68	523.52	956.93	961.40	953.79
R1	418.26	413.45	407.93	1001.70	1006.34	1036.68	1398.49	1387.93	1395.39
R2	189.35	186.32	195.06	806.66	799.23	826.28	1272.81	1272.97	1290.91
RC1	442.88	447.62	446.73	1249.09	1253.26	1272.63	1597.93	1606.71	1635.91
RC2	368.49	369.45	375.94	1257.42	1261.28	1285.04	1833.07	1847.33	1907.96
Medium Fleet Size, a=0.5									
Class	Focus on Tardiness, w=0.1			Focus on both costs, w=0.5			Focus on Traveling, w=0.9		
	VNS	LNS	TS	VNS	LNS	TS	VNS	LNS	TS
C1	2185.00	2201.17	2265.07	2309.91	2310.95	2330.99	1912.01	1921.68	1918.90
C2	108.26	108.48	107.58	539.31	541.68	529.88	951.01	957.11	974.09
R1	1485.68	1497.27	1479.78	1431.19	1436.15	1439.08	1391.34	1401.08	1397.73
R2	371.16	373.75	371.82	873.04	882.44	885.05	1265.23	1275.28	1270.98
RC1	1366.80	1367.79	1374.72	1508.49	1544.69	1549.96	1584.38	1602.04	1615.85
RC2	1082.31	1087.83	1106.36	1535.63	1572.54	1570.12	1837.25	1844.29	1837.64

Small Fleet Size, a=0.7									
Class	Focus on Tardiness, w=0.1			Focus on both costs, w=0.5			Focus on Traveling, w=0.9		
	VNS	LNS	TS	VNS	LNS	TS	VNS	LNS	TS
C1	5347.30	5290.05	5411.78	3975.90	3986.07	4036.76	2142.91	2141.38	2146.42
C2	106.75	106.75	109.98	539.25	539.37	558.94	954.67	958.85	960.03
R1	2822.64	2798.82	2859.09	2223.69	2249.90	2325.22	1447.89	1437.73	1464.98
R2	833.24	819.46	843.15	1163.27	1146.93	1160.85	1303.51	1301.17	1303.20
RC1	2933.30	2945.85	2980.39	2127.50	2146.84	2145.31	1602.23	1590.91	1618.86

Table 1 is average of average solution cost of each class. The table shows the solution on three different fleet sizes; i.e., big, medium and small fleet size in which each has been focusing on different cost solution in the objective function; focus on tardiness, traveling and both costs equally.

For big fleet case, the initial available vehicles are more than needed. As mentioned in our problem description before, minimize the vehicle number is not the aim. In fact we are trying to maximize the usage of available vehicles through the fleet size-cases that we had designed which might improve the solution cost. From the Table 1, it can be seen that the more constrained the problem is, contributes the cost higher.

As expected, the problem sets with the wider time between dates generate the least cost than the sets with the narrower time. For example, in the small fleet case of heuristic VNS, the average cost 5347.30 and 106.75 are obtained by C1 and C2 respectively. The same situation happens in other heuristics and problem classes. However, different circumstances occur in the problem sets of RC1 and RC2. RC1 generates the least cost than RC2 when the problem focusing on traveling and, both costs equally.

For our experiment, it is empirically found that solutions generated in the big fleet case produce the least costs. This can be seen in all type of weights, *w* in which focusing on tardiness in the objective function yield the best solution. However, unlike situation in other two fleet cases i.e. medium and small-fleet sizes, focusing on traveling, i.e. *w = 0.9* gives the least cost to each of the problem classes compared to other weights.

Logically, by fully utilized the resources without incurring the vehicle cost might benefit to improve the solution cost especially the tardiness. However, with the limited vehicles we have, focusing on distance might be the best strategy. In overall, solution in class C2 does not show any significant difference even though different fleet sizes have been used. This is due to the nature of instances which has been classified as clustered and having wider time windows.

The next Table 2 gives the clear picture of the performance of each algorithm. The relative percentage deviation was calculated based on the average solution of all classes as in the previous Table 1.

Table 2 Comparison of the heuristics.

Fleet Sizes	**Weights**	**Methods**		
		VNS	**LNS**	**TS**
Big-Fleet *a=0.3*	*0.1*	0.00	0.22	0.04
	0.5	0.05	0.00	0.56
	0.9	0.00	0.19	1.98
Medium-Fleet, *a=0.5*	*0.1*	0.00	0.56	1.61
	0.5	0.00	1.11	1.31
	0.9	0.00	0.67	0.83
Small-Fleet *a=0.7*	*0.1*	0.30	0.00	1.97
	0.5	0.01	0.00	1.42
	0.9	0.11	0.00	0.79
Average Percentage Deviation.		**0.05**	0.31	1.17

Table 2 shows an average of percentage deviation on the average cost solution for all six classes. Clearly it can be seen that VNS is the most effective heuristic, solving the problems with 100 customers to the near optima and achieving the least average solution cost in 5 out of 9 solutions. It followed by LNS which gives the least cost in the rest of 4 solutions. Note here, we set all the approached heuristics producing the solution on the same limited CPU running time.

4. CONCLUSION

The study presents the three most efficient heuristics to solve the vehicle routing problem with time constraints. The problem has been categorized as non-classical VRP in which departure time of each vehicle is depends on the maximum release date of customer's order scheduled in the particular vehicle. The proposed algorithms have been tested using modified 56' Solomon problems with 100 customers.

The computational results showed that VNS has been found as the best method to solve the problem studied. Results obtained shown that the implementation of VNS which takes a randomized search is work efficiently.

ACKNOWLEDGEMENTS

A special thanks to University of Southampton for the opportunity to carry out the research and to UniversitiTeknologi Malaysia (UTM) and Ministry of Higher Education (MOHE) for financial support. This project was supported by Research University Grant [Vote Number: 09J67] initiated by UTM and MOHE.

REFERENCES

[1] Kumar, S.N., and Panneerselvam, R, *A Survey on Vehicle Routing Problem and Its Variants*, Intelligent Information Management(2012), 4, 66-74.

[2] BI.Kim, S.Kim, and S.Sahoo, *Waste collection vehicle route problem with time window*. Computers & Operational Research 33 (2006), pp.3624-3642.

[3] N.Balakrishnan, *Simple heuristics for the vehicle routeing problem with soft time windows*. Journal of the Operational Research Society 44, **3** (1993), pp.279-287.

[4] J.C.Chen, H. W. H. C. C., and C.Chen, *Hybrid meta-heuristics for vehicle routing problem with time window constraints,*In Service System and Service Management, 2009. ICSSSM`09. 6th International Conference Proceedings, International Conference on (2009), IEEE, pp. 766-771.

[5] H.A.Min,*Amultiobjective vehicle routing problem with soft time windows: the case of a public library distribution system.* Socio-Economic Planning Sciences 25, 3 (1991), pp.179-188.

[6] L.Bodin, B.Golden, A. Assad, and M.Ball, *Routing and scheduling of vehicles and crews: The state of the art.* COMP. & OPER. RES. 10, 2 (1983), pp.63-211.

[7] M.Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints.* Operations Research (1987), pp.254-265.

[8] J.Cordeau, M.Gendreau, G.Laporte, J.Potvin,and F.Semet, *A guide to vehicle routing heuristics*. Journal of the Operational Research Society (2002), pp.512-522.

[9] O.Braysy andM.Gendreau,*Vehicle routing problem with time windows. Part I: Route construction and local search algorithms*. Transportation Science 39, 1 (2005), pp.104-118.

[10] O.Braysy andM.Gendreau,*Vehicle routing problem with time windows. Part II: Metaheuristics.* Transportation Science 39, 1 (2005), pp.119-139.

[11] F.Liu, and S.Shen,*An overview of a heuristic for vehicle routing problem with time windows.* Computers & industrial engineering 37, 1-2 (1999), pp.331-334.

[12] Figliozzi, M.A. *The time dependent vehicle routing problem with time windows: Benchmarks problems, an efficient solution algorithm and solution characteristics*, Transportation Research Part E 48 (2012), 616-636.

[13] El-Sherbeny, N. *Vehicle routing with time windows: An Overview of exact, heuristic and metaheuristic methods*. Journal of King Sad University (Science) (2010), 121-131.

[14] Taner, F., Galic, A. and Caric, T. *Solving practical vehicle routing problem with time windows using metaheuristic algorithms*, Traffic&Transportation, (2012), 24, 343-351.

[15] Minocha, B. and Tripathi, S. *Two phase algorithm for solving VRPTW problem*. International Journal of Artificial Intelligence and Expert Systems (IJAE)(2013), 4, 1-16.

[16] Mladenovic, N., and Hansen, P. *Variable neighborhood search*. Computers & Operations Research 24, 11(1997), 1097-1100.

[17] L. Hong, *An improved LNS algorithm for real-time vehicle routing problem with time windows*. Computers & Operations Research 39 (2012), 151-163.

[18] Stenger A., Vigo D., Enz S., Schwind M., *An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping*. Transportation Science 47 (1), 2013, 64-80.

[19] Korsvik, J. E., Fagerholt, K., Laporte, G., *A large neighbourhood search heuristic for ship routing and scheduling with split loads*. Computers & Operations Research 38 (2011), 474-483.

[20] Kovacs, A. A., Parragh S. N., and Hartl R. F., *A template based adaptive large neighborhood search for the consistent vehicle routing problem*. Networks 63 (1), 2014, 60-81.

[21] Glover, F., and Laguna, M., *Tabu search*. 1997.

[22] Marinakis, Y., and Migdalas, A., *Heuristic solutions of vehicle routing problem in supply chain management*. Series on Applied Mathematics-World Scientific Publishing Company-14 (2002), 205-236.

[23] Cordeau, J., and Laporte, G., *Tabu search heuristics for the vehicle routing problem*. Metaheuristic Optimization via Memory and Evolution (2005), 145-163.

[24] Moccia, L., Cordeau, J. F., and Laporte G., *An incremental tabu search heuristic for the generalized vehicle routing problem with time windows*. Journal of the Operational Research Society 63, (2012), 232-244.

[25] R. Liu, X. Xie, Augusto, V., Rodriguez, C., *Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care*. European Journal of Operational Research 230 (2013), 475-486.

[26] Ceschia, S., Gaspero, L. D., Schaerf A., *Tabu search techniques for the heterogeneous vehicle routing problem with time windows and carrier-dependent costs*. Journal of Scheduling 14, 6 (2011), 601-615.

[27] H. Lau, M. Sim, and K. Teo, *Vehicle routing problem with time windows and a limited number of vehicles*. European Journal of Operational Research 148, 3 (2003), 559-569.

[28] A. Lim, and F. Wang, *A smoothed dynamic tabu search embedded GRASP for m-VRPTW.* In Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on (2005), IEEE, pp. 704-708.

[29] A. Lim, X. Zhang, *A two-stage heuristic for the vehicle routing problem with time windows and a limited number of vehicles*. In Proceedings of the 38th Hawaii International Conference on System Sciences (2005).

[30] X. Wang, C. Xu, X. Hu, *Genetic algorithm for vehicle routing problem with time windows and a limited number of vehicles*. In Management Science and Engineering, 2008. ICMSE 2008.15th Annual Conference Proceedings. International Conference on (2008), IEEE, pp. 128-133.

[31] Y. Park, *A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines*. International Journal of Production Economics 73, 2 (2011), 175-188.

[32] K. H. Kang, B. K. Lee, Y. H. Lee, Y. H. Lee, *A heuristic for the vehicle routing problem with due times*. Computers & Industrial engineering 54 (2008), 421-431.

[33] X.Wang, C.Xu, and X.Hu, *Genetic algorithm for vehicle routing problem with time windows and a limited number of vehicles*. In Management Science and Engineering, 2008. ICMSE 2008. 15th Annual Conference Proceedings, International Conference on (2008), IEEE, pp. 128-133.