International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 3 Issue: 9

ISSN: 2321-8169
5508 - 5513

Mixed Integer Programming Model for open Vehicle Routing Problem with Fleet and driver Scheduling Considering Delivery and Pick-Up Simultaneously

Hardi Tambunan
Departement of Mathematics
Quality University
Medan, Indonesia

Herman Mawengkang
Departement of Mathematics
University of Suamtera Utara
Medan, Indonesia
hmawengkang@yahoo.com

Abstract—Vehicle Routing Problem (VRP) is a key element of many logistic systems which involve routing and scheduling of vehicles from a depot to a set of customers node. This is a combinatorial optimization problem with the objective to find an optimal set of routes used by a fleet of vehicles to serve a set of customers It is required that these vehicles return to the depot after serving customers' demand. This paper investigates a variant of VRP, in which the vehicles do not need to return to the depot, called open vehicle routing problem (OVRP). The problem incorporates time windows, fleet and driver scheduling, pick-up and delivery in the planning horizon. The goal is to schedule the deliveries according to feasible combinations of delivery days and to determine the scheduling of fleet and driver and routing policies of the vehicles. The objective is to minimize the sum of the costs of all routes over the planning horizon. We model the problem as a linear mixed integer program. We develop a combination of heuristics and exact method for solving the model.

Keywords- *Logistics, Integer programming, Heuristics, Neighborhood search*

*****

I. INTRODUCTION

Vehicle Routing Problem (VRP) is one of the important issues that exist in transportation system. This is a well known combinatorial optimization problem which consists of a customer population with deterministic demands, and a central depot which acts as the base of a homogeneous fleet of vehicles. The objective is to design a set of Hamiltonian cycles (vehicle routes) starting and terminating at the central depot, such that the demand of customers is totally satisfied, each customer is visited once by a single vehicle, the total demand of the customers assigned to a route does not exceed vehicle capacity, and the overall travel cost, taking into account various operational constraints. VRP was first introduced by [8]. Since then many researchers have been working in this area to discover new methodologies in selecting the best routes in order to find the better solutions. There are a number of survey can be found in literature for VRP, such as ([6], [5], [1] , [3]), and books ([2], [7]).

Mathematically, *VRP* can be defined as follows: vehicles with a fixed capacity Q must deliver order quantities q_i ($i = 1, \dots, n$) of goods to n customers from a single depot (i = 0). Knowing the distance d_{ij} between customers i and j ($i, j = 1, \dots, n$), the objective of the problem is to minimize the total distance traveled by the vehicles in a way that only one vehicle handles the deliveries for a given customer and the total quantity of goods that a single vehicle delivers is not larger than Q [18].

In some cases, particularly, when the business firms do not own a vehicle fleet, or their private fleet is inadequate for fully satisfying customer demand, distribution services would be carried out by external contractors, such as a hired vehicle fleet. Therefore vehicles are not required to return to the central depot after their deliveries have been satisfied. This distribution model is referred to as an open vehicle routing problem (OVRP). Open vehicle routing problem is an expansion problem of the classic vehicle routing problem. The most significant difference between OVRP and VRP is that in the OVRP, vehicles do not return to the original depot after servicing the last customer on the route, or if they are required, they return by traveling the same route back. Open vehicle routing problem is a key step of logistics optimization and the indispensable part of the ecommerce activities.

Following the VRP which belongs to an NP-hard problem the OVRP is also an NP-hard; therefore to deal with OVRP instances of practical size, researchers have focused their interest on the development of effective heuristic and metaheuristic solution approaches.

The article of [13] which classifies the features encountered in practical routing problems was the first to distinguish between closed trips traveled by private vehicles, and open trips assigned to common carrier vehicles. The first solution approach for the OVRP is due to [12]. Their paper deals with a practical routing problem faced by the airplane fleet of FedEx. In specific, airplanes layover at the end of their delivery routes, to later perform their pick-up trips. These delivery routes can be seen as an application of the OVRP, in the sense that airplanes do not return to the depot. Their solution approach is a variant of the [21] algorithm adapted to the examined problem.

Following the VRP which belongs to an NP-hard problem the OVRP is also an NP-hard; therefore to deal with OVRP instances of practical size, researchers have focused their interest on the development of effective heuristic and metaheuristic solution approaches. Reference [14] address a heuristic method based on a minimum spanning tree combined with a penalization procedure for solving OVRP. Reference [11] presents a tabu search procedure which makes use of customer insertion and swap local search operators. Reference [15, 16] have presented studies of meta-heuristics on the OVRP, which belong to the threshold accepting category of algorithms. The first one [15] proposes an annealing based method that utilizes a backtracking policy of the threshold value when no acceptances of feasible solutions occur during the search process, whereas the second one [16] presents a single-parameter metaheuristic method that exploits a list of threshold values to intelligently guide an advanced local search method.

Reference [17] have also published an adaptive memory approach for the OVRP. Their approach involves a pool of routes that belong to the highest quality solutions encountered through the search process. Sequences of customers are extracted from the adaptive memory to form new partial solutions later to be improved by a tabu search procedure. The routes of these improved solutions are used to update the adaptive memory contents, forming in this way a cyclic algorithm. Note that the aforementioned three works aim at solely minimizing the total distance of the open routes, disregarding the required fleet size.

Reference [9,10] propose a metaheuristic framework which constructs an initial OVRP solution via a farthest-first heuristic. This solution is then improved by a tabu search method which employs the well-known relocation, swap, and 2-opt operators. reference [18] have dealt with the OVRP by developing a local search metaheuristic algorithm which uses the concept of record-to-record travel [19]. In their work, they introduce eight large-scale OVRP instances which have served as a comparison basis for the effectiveness of recent OVRP methodologies. These works include the general routing heuristic of [20] which has been effectively applied to the OVRP variant. Their approach involves the application of the adaptive large neighborhood search framework.

Hybrid genetic algorithm was used by [22] to solve a variant of OVRP which involve single and mixed fleet strategy. Reference [23] propose a hybrid ant colony metaheuristic approach for OVRP. They present a new transition rule, an efficient candidate list, several effective local search techniques and a new pheromone updating rule in a way to achieve better solution.

From literature survey mentioned before most of the research are on how to solve OVRP using metaheuristic. In reality there some variant for this OVRP This paper concerns with a comprehensive model for the variant of OVRP which incorporated time windows, fleet and driver scheduling, pick-up and delivery (OVRFDPDTWP) . The basic framework of the vehicle routing part can be viewed as a Heterogeneous Vehicle Routing Problem with Time Windows (HVRPTW) in which a limited number of heterogeneous vehicles, characterized by different capacities are available and the customers have a specified time windows for services. We propose a mixed integer programming formulation to model the problem. A feasible neighbourhood heuristic search is addressed to get the integer feasible solution after solving the continuous model of the problem.

II. MATHEMATICAL FORMULATION OF OVRDPDTWP

Using graph, OVRP can be defined as follows. Let a graph G = (V, E), where $V = \{v_0, v_1, \ldots, v_n\}$ is the vertex set and $E = \{(v_i, v_j)\colon v_i, v_j \in V,\ i \neq j,\ j \neq 0\}$ is the arc set. Vertex v_0 represents the central depot where a fleetof vehicles is located, each of them with maximum carrying load equal to Q. The remaining n vertices of $V \setminus \{v_0\}$ represent the customer set. With each customer vertex is associated a non-negative known demand q_i, whereas with each arc $(v_i, v_j) \in E$ is associated a cost c_{ij} which corresponds to the cost (travel time, distance) for transiting from v_i to v_j. As with most previous OVRP approaches, we consider that the cost matrix is obtained by calculating the Euclidean distances between vertex pairs, so that $c_{ij} = c_{ji}$ $(0 < i, j \leq n, i \neq j)$. The objective of the problem is to design the set of Hamiltonian paths to serve all customers such that: the number of vehicles is minimized, as well as to minimize the total cost of the generated paths. There are some restrictions which must be satisfied, such as, every path originates from the central depot v_0, each customer vertex is assigned to a single path, and the total demand of the customer set assigned to a single path does not exceed the maximum carrying load Q of the vehicles (capacity constraint).

To formulate the model, firstly we denote T as the planning horizon and D as the set of drivers. The set of workdays for driver $l \in D$ is denoted by $T_l \subseteq T$. The start working time and latest ending time for driver $l \in D$ on day $t \in T$ are given by g_l^t and h_l^t , respectively. For each driver $l \in D$, let H denote the maximum weekly working duration. We denote the maximum elapsed driving time without break by F and the duration of a break by G .

Let K denote the set of vehicles. For each vehicle $k \in K$, let Q_k and P_k denote the capacity in weight and in volume, respectively. We assume the number of vehicles equals to the number of drivers. Denote the set of n customers (/nodes) by $N = \{1, 2, \ldots, n\}$. Denote the depot by $\{0, n+1\}$. Each vehicle starts from $\{0\}$ and terminates at $\{n+1\}$. Each customer $i \in N$ specifies a set of days to be visited, denoted by $T_i \subseteq T$. On each day $t \in T_i$, customer $i \in N$ requests service with demand of q_i^t in weight and p_i^t in volume, service duration d_i^t and time window $[a_i, b_i]$. Note that, for the depot $i \in \{0, n+1\}$ on day t, we set $q_i^t = p_i^t = d_i^t = 0$. Denote the set of preferable vehicles for visiting customer i by K_i ($K_i \in K$) and the extra service time per pallet by e if a customer is not visited by a preferable vehicle. The travel time between customer i and j is given by c_{ij} . Denote the cost coefficients of the travel time of the internal drivers by A and the working duration of the external drivers by B.

We define binary variable x_{ijk}^t to be 1 if vehicle k travels from node i to j on day t, binary variable w_i^t to be 1 if customer i is not visited by a preferred vehicle on day t. Variable v_{ik}^t is the time that vehicle k visits node i on day t. Binary variable z_{ik}^t indicates whether vehicle k takes a break after serving customer i on day t. Variable u_{ik}^t is the elapsed driving time for vehicle k at customer i after the previous break on day t. Binary variable y_{lk}^{t} is set to 1 if vehicle k is assigned to driver l on day t. Variables r_l^t and s_l^t are the total working duration and the total travel time for driver l on day t, respectively.

Notations used are defined as follows.

Set:

T	The set of workdays in the planning horizon,
D	The set of drivers $D = D_I \cup D_E$,
T_l	The set of workdays for driver $l \in D$,
K	The set of vehicles,
N	The set of customers,
N_0	The set of customers and depot $N_0 = \{0, n + 1\} \cup N$,
K_i	The set of preferable vehicles for customer $i \in N$,
T_i	The set of days on which customer $i \in N$ orders,

Parameter:

Q_k	The weight capacity of vehicle $k \in K$,
P_k	The volume capacity of vehicle $k \in K$,
c_{ij}	The travel time from node $i \in N_0$ to node $j \in N_0$,
$[a_i, b_i]$	The earliest and the latest visit time at node $i \in N_0$,
d_i^t	The service time of node $i \in N_0$ on day $t \in T_i$,
q_i^t	The weight demand of node $i \in N_0$ on day $t \in T_i$,
p_i^t	The volume demand of node $i \in N_0$ on day $t \in T_i$,
e	The extra service time per pallet when a non-preferable vehicle is used,
$[g_l^t, h_l^t]$	The start time and the latest ending time of driver $l \in D$ on day $t \in T$,
α_i^t	Pick up quantity for customer i on day $t \in T_i$,
β_i^t	Delivery quantity for customer I on day $t \in T_i$,
H	The maximum working duration for each internal driver over the planning horizon,
F	The maximum elapsed driving time without break,
G	The duration of the break for drivers,
A	The cost factor on the total travel time

Variables:

x_{ilk}^t	Binary variable indicating whether vehicle $k \in K$ travels from node $i \in N_0$ to $j \in N_0$ on day $t \in T$,
w_i^t	Binary variable indicating whether customer $i \in N_0$ is visited by a non-preferable vehicle on day $t \in T$,
v_{ik}^t	The time at which vehicle $k \in K$ starts service at node $i \in N_0$ on day $t \in T$,
z_{ik}^t	Binary variable indicating whether vehicle $k \in K$ takes break after serving node $i \in N_0$ on day $t \in T$,
u_{ik}^t	The elapsed driving time of vehicle $k \in K$ at node $i \in N_0$ after the previous break on day $t \in T$,
y_{lk}^t	Binary variable indicating whether vehicle $k \in K$ is assigned to driver $l \in D$ on day $t \in T$,
r_l^t	The total working duration of driver $l \in D$ on day $t \in T$,
s_l^t	The total travel distance of driver $l \in D$ on day $t \in T$,
θ_{jk}^t	Number of pick up demand of customer j served by vehicle $k \in K$ on day $t \in T$
σ_{jk}^t	Number of delivery demands of customer j served by vehicle $k \in K$ on day $t \in T$

The problem can be presented as a mixed integer linear programming model.

The objective of the problem is to minimize cost. Firstly, we sum up the total travel time in the planning horizon, and then we multiply the result with cost factor A. Therefore the objective can be expressed as follows.

Minimize

$$A\left(\sum_{j\in N}\sum_{k\in K}\sum_{t\in T} c_{0jk}^t x_{0jk}^t + \sum_{i\in N_0}\sum_{j\in N}\sum_{k\in K}\sum_{t\in T} c_{ijk}^t\right) \qquad (1)$$

Subject to:

$$\sum_{i\in N_0} x_{0i}^t = d \qquad \forall\ d \in D,\ t \in T \qquad (2)$$

$$\sum_{j\in N} x^t_{ji} = 1 \qquad \forall i\in N_0, t\in T \qquad (3)$$

$$\sum_{i\in N_0} x^t_{ij} = 1 \qquad \forall j\in N, t\in T \qquad (4)$$

$$\sum_{k\in K\setminus K_i}\sum_{j\in N_0} x^t_{ijk} = w^t_i \qquad \forall i\in N, t\in T \qquad (5)$$

$$\sum_{i\in N}\sum_{j\in N_0} q^t_i x^t_{ijk} \le Q_k \qquad k\in K, t\in T \qquad (6)$$

$$\sum_{i\in N}\sum_{j\in N_0} p^t_i x^t_{ijk} \le P_k \qquad k\in K, t\in T \qquad (7)$$

$$u^t_{jk} \ge u^t_{ik} + c_{ij} - M(1-x^t_{ijk}) - Mz^t_{ik}$$
$$\forall i, j \in N_0,\ k\in K,\ t\in T \qquad (8)$$

$$u^t_{jk} \ge c_{ij} - M(1-x^t_{ijk}) \quad \forall i, j\in N,\ k\in K,\ t\in T \qquad (9)$$

$$u^t_{ik} + \sum_{j\in N_0} c_{ij}x^t_{ijk} - F \le Mz^t_{ik}$$
$$\forall i\in N_0,\ k\in K,\ t\in T \qquad (10)$$

$$v^t_{jk} \ge v^t_{ik} + d^t_i + e\cdot p^t_i\cdot w^t_j + c_{ij} + G\cdot z^t_{ik} - M(1-x^t_{ijk})$$
$$\forall i, j\in N_0,\ k\in K,\ t\in T \qquad (11)$$

$$b_i \ge v^t_{ik} \ge a_i \qquad \forall i\in N,\ k\in K,\ t\in T \qquad (12)$$

$$v^t_{0k} \ge \sum_{l\in D}(g^t_l\cdot y^t_{lk}) \qquad \forall k\in K,\ t\in T \qquad (13)$$

$$v^t_{n+1,k} \le \sum_{l\in D}(h^t_l\cdot y^t_{lk}) \qquad \forall k\in K,\ t\in T \qquad (14)$$

$$s^t_l \ge \sum_{i\in N_0}\sum_{j\in N_0} c_{ij}x^t_{ijk} - M(1-y^t_{lk})$$
$$\forall l\in D, k\in K, t\in T \qquad (15)$$

$$r^t_l \ge v^t_{n+1,k} - g^t_l - M(1-y^t_{lk})$$
$$l\in D, k\in K, t\in T \qquad (16)$$

$$\sum_{t\in T_l} r^t_l \le H \qquad l\in D \qquad (17)$$

$$\sum_{k\in K}\theta^t_{jk} = \alpha^t_j \qquad \forall j\in N, t\in T \qquad (18)$$

$$\sum_{k\in K}\sigma^t_{jk} = \beta^t_j \qquad \forall j\in N, t\in T \qquad (19)$$

$$x^t_{ijk}, w^t_i, z^t_{ik}, y^t_{lk} \in \{0,1\}$$
$$\forall i, j\in N_0,\ l\in D,\ k\in K,\ t\in T \qquad (20)$$

$$v^t_{ik}, u^t_{ik}, r^t_l, s^t_l \ge 0$$
$$\forall i, j\in N_0,\ l\in D,\ k\in K,\ t\in T \qquad (21)$$

$$\theta^t_{jk}, \sigma^t_{jk} \in \{0,1,2,\ldots\}$$
$$\forall j\in N, k\in K, t\in T \qquad (22)$$

As this is an OVRP, the vehicles used are only dispatched from depot. Constraint (2) is to make sure that the number of vehicle dispatches from depot should be the same as the number of driver. Constraint (3) expresses that at each node, beside depot, there should be exactly one entering arc coming from a customer node or from the depot. The other way around is expressed in Constraint (4). Constraints (5) define whether each customer is visited by a preferable vehicle. Constraints (6-7) guarantee that the vehicle capacities are respected in both weight and volume.

Constraints (8-9) define the elapsed driving time. More specifically, for the vehicle (k) travelling from customer i to j on day t, the elapsed driving time at j equals the elapsed driving time at i plus the driving time from i to j (i.e., $u^t_{jk} \ge u^t_{ik} + c_{ij}$) if the vehicle does not take a break at customer i (i.e., z^t_{ik} = 0); Otherwise, if the vehicle takes a break at customer i (i.e., z^t_{ik} = 1), the elapsed driving time at j will be constrained by (9) which make sure it is greater than or equal to the travel time between i and j (i.e., $u^t_{jk} \ge c_{ij}$). Constraints (10) guarantee that the elapsed driving time never exceeds an upper limit F by imposing a break at customer i (i.e., z^t_{ik} = 1) if driving from customer i to its successor results in a elapsed driving time greater than F.

Constraints (11) determine the time to start the service at each customer. If j is visited immediately after i, the time v^t_{jk} to start the service at j should be greater than or equal to the service starting time v^t_{ik} at i plus its service duration d^t_i, the extra service time $e\cdot p^t_i$ if i is visited by an inappropriate vehicle (w^t_j = 1), the travel time between the two customers cij , and the break time G if the driver takes a break after serving I (i.e., z^t_{ik} = 1). Constraints (12) make sure the services start within the customers' time window.

Constraints (13-14) ensure that the starting time and ending time of each route must lie between the start working time and latest ending time of the assigned driver. Constraints (15) calculate the total travel time for each internal driver. Constraints (16) define the working duration for each driver on every workday, which equals the time the driver returns to the depot minus the time he/she starts work. Constraints (17) make sure that the internal drivers work for no more than a maximum weekly working duration, referred to as 37 week-hour constraints. Constraints (18 – 19) define the pick up and delivery for each customer. Constraints (20-21) define the binary and positive variables used in this formulation.

The problem expressed as a mixed integer programming model contains a large number of variables.

III. NEIGHBORHOOD SEARCH

It should be noted that, generally, in integer programming the reduced gradient vector, which is normally used to detect an optimality condition, is not available, even though the problems are convex. Thus we need to impose a certain condition for the local testing search procedure in order to assure that we have obtained the "best" suboptimal integer feasible solution.

Further in [4] has proposed a quantity test to replace the pricing test for optimality in the integer programming problem. The test is conducted by a search through the neighbors of a proposed feasible point to see whether a nearby point is also feasible and yields an improvement to the objective function.

Let $[\beta]_k$ be an integer point belongs to a finite set of neighborhood $N([\beta]_k)$ We define a neighborhood system associated with $[\beta]_k$, that is, if such an integer point satisfies the following two requirements

1. if $[\beta]_j \in N([\beta]_k)$ then $[\beta]_k \in [\beta]_j$, $j \neq k$.
2. $N([\beta]_k) = [\beta]_k + N(0)$

With respect to the neighborhood system mentioned above, the proposed integerizing strategy can be described as follows.

Given a non-integer component, x_k, of an optimal vector, x_B. The adjacent points of x_k, being considered are $[x_k]$ dan $[x_k] + 1$. If one of these points satisfies the constraints and yields a minimum deterioration of the optimal objective value we move to another component, if not we have integer-feasible solution.

Let $[x_k]$ be the integer feasible point which satisfies the above conditions. We could then say if $[x_k] + 1 \in N([x_k])$ implies that the point $[x_k] + 1$ is either infeasible or yields an inferior value to the objective function obtained with respect to $[x_k]$. In this case $[x_k]$ is said to be an "optimal" integer feasible solution to the integer programming problem. Obviously, in our case, a neigbourhood search is conducted through proposed feasible points such that the integer feasible solution would be at the least distance from the optimal continuous solution.

IV. THE ALGORITHM

We combine exact method and heuristics for solving this large scale mixed integer programming problem. Firstly, we solve the linear programming part after relaxing the integer restriction. Then, we using the following heuristics for searching a suboptimal but integer-feasible solution.

Let

$$x = [x] + f, \quad 0 \le f \le 1$$

be the (continuous) solution of the relaxed problem, $[x]$ is the integer component of non-integer variable x and f is the fractional component.

Stage 1.

Step 1. Get row i^* the smallest integer infeasibility, such that

$$\delta_{i^*} = \min\{f_i, 1 - f_i\}$$

(This choice is motivated by the desire for minimal deterioration in the objective function, and clearly corresponds to the integer basic with smallest integer infeasibility).

Step 2. Do a pricing operation

$$v_{i^*}^T = e_{i^*}^T B^{-1}$$

Step 3. Calculate $\sigma_{ij} = v_{i^*}^T \alpha_j$

With corresponds to

$$\min_j \left\{ \left| \frac{d_j}{\alpha_{ij}} \right| \right\}$$

Calculate the maximum movement of nonbasic j at lower bound and upper bound.

Otherwise go to next non-integer nonbasic or superbasic j (if available). Eventually the column j^* is to be increased form LB or decreased from UB. If none go to next i^*.

Step 4. Solve $B\alpha_{j^*} = \alpha_{j^*}$ for α_{j^*}

Step 5. Do ratio test for the basic variables in order to stay feasible due to the releasing of nonbasic j^* from its bounds.

Step 6. Exchange basis

Step 7. If row $i^* = \{\varnothing\}$ go to Stage 2, otherwise

Repeat from step 1.

Stage 2. Pass1 : adjust integer infeasible superbasics by fractional steps to reach complete integer feasibility.

Pass2 : adjust integer feasible superbasics. The objective of this phase is to conduct a highly localized neighborhood search to verify local optimality.

V. CONCLUSIONS

This paper is intended to develop efficient technique for solving one of the most economic importance problems in optimizing transportation and distribution systems. The aim of this paper is to develop a model of open vehicle routing with Time Windows, Fleet and Driver Scheduling, Pick-up and Delivery Problem This problem has additional constraint which is the limitation in the number of vehicles. The proposed algorithm employs nearest neighbor heuristic algorithm for solving the model.

REFERENCES

[1] B. L. Golden, A. A. Assad, and E. A. Wasil, "Routing Vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries". In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 245-286. SIAM, Philadelphia, PA, 2002.

[2] B. L. Golden, S. Raghavan, and E. A. Wasil, "The vehicle routing problem: latest advances and new challenges". Springer, 2008.

[3] G. Laporte and F. Sement. Classical Heuristics for the Vehicle Routing Problem. In Toth, P. and Vigo, D., editors, *The vehicle Routing Problem*, SIAM Monographs on Discrete Matheamtics and Applications, pages 109-128. SIAM, Philadelphia, PA, 2002.

[4] H. E Scarf., *Testing for optimality in the absence of convexity* in Walter P Heller, Ross M Starr and David A Starrett (Eds) (Cambridge University Press) pp 117-134, 1986.
[5] J.-F. Cordeau, G. Laporte, M.W.F. Savelsbergh, and D. Vigo. Vehivle Routing. In Barnhart, C. and Laporte, G., edotirs, *Transportation,* handbooks in Operation Research and Management Science, Pages 367-428. North-Holland, Amsterdam, 2007.
[6] O. Bräysy and M. Gendreau. "Vehicle Routing Problem with Times Windows, Part II: Metaheuristics". *Transportation Science,* 39(1): 104-118, 2005.
[7] P. Toth and D. Vigo. "The Vehicle Routing Problem Society for Industrial and Applied Matheamtics", Philadelpiha, PA, 2002.
[8] G.B. Dantzig and J.H. Ramser, "The truck dispatching problem", Management Science, Vol. 6, 1959, 80.
[9] Fu Z, Eglese RW, Li LYO. A new tabu search heuristic for the open vehicle routing problem. J Oper Res Soc 2005;56:267-74.
[10] Fu Z, Eglese RW ,Li LYO. Corrigendum. A new tabu search heuristic for the open vehicle routing problem. J Oper Res Soc 2006;57:1018.
[11] J. Brandão . A tabu search algorithm for the open vehicle routing problem. Eur J Oper Res 2004; 157(3): 552-64.
[12] L. Bodin, Golden B, Assad A, Ball M. Routing and scheduling of vehicles and crews : The state of the art. Comput Oper Res 1983;10(2):63-211.
[13] L. Schrage. Formulation and structure of more complex/realistic routing and scheduling problems. Networks 1981;11(2):229-32.
[14] D. Sariklis, Powell S. A heuristic method for the open vehicle routing problem. *J Oper Res Soc* 2000;51(5):564-573.
[15] C.D. Tarantilis, Ioannou G, Kiranoudis CT, Prastacos GP. A threshold accepting approach to the open vehicle routing problem. RAIRO 2004;38:345-60.
[16] C.D. Tarantilis, Kiranoudis CT, Ioannou G, Prastacos GP. Solving the open vehicle routing problem via a single parameter metaheuristic algorithm. *J Oper Res Soc* 2005;56:588-96.
[17] C.D. Tarantilis, Diakoulaki D, Kiranoudis CT. Combination of geographical information system and effective routing algorithms for real life distribution operations. *Eur J Oper Res* 2004;152:437-53.
[18] Li F, Golden B, Wasil E. The open vehicle routing problem: Algorithms, largescale test problems, and computational results. *Comput Oper Res* 2004;34(10):2918-30.
[19] Dueck G. New optimization heuristics: the great deluge algorithm and the recordto-record travel. *J Comput Phys* 1993;104:86-92.
[20] D. Pisinger, Ropke S. A general heuristic for vehicle routing problems. Comput Oper Res 2007;34(8):2403-35. Math Model Algor 2004;3(2):167-78.
[21] G. Clarke, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Oper Res 1964; 12: 568-89.
[22] C. Ren. Research on Single and Mixed Fleet Strategy for Open Vehicle Routing Problem. Journal of Software 2011;6(10):2076-2081.
[23] M. Sedighpour, V. Ahmadi, M. Yousefikhoshbakht, F. Didehvar, F. Rahmati. Solving the Open Vehicle Routing Problem by a Hybrid Ant Colony Optimization. Kuwait J. Sci. 2014;41(3):139-162