Mixed Integer Programming Model for Split Delivery Vehicle Routing Problem with Fleet and Driver Scheduling

Arnita
State University of Medan,
Medan, Indonesia

Herman Mawengkang
Department of Mathematics
University of Sumatera Utara, Medan, Indonesia

Abstract: Vehicle Routing Problem (VRP) is a key element of many transportation systems which involve routing of fleet of vehicles from a depot to a set of customers node. It is required that these vehicles return to the depot after serving customers' demand. This paper investigates a relaxed version of VRP, in which the number of visits to the customer is not restricted to be at most one. The relaxed version is called split delivery VRP. The problem incorporates time windows, fleet and driver scheduling 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 total 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: *Routing, Scheduling, Transportation, Combined method*

I. Introduction

In logistic system it is necessarily to design the optimal routes for a set of vehicle involved with given capacity in such a way that could satisfy customers' demand. Vehicle Routing Problem (VRP) is one of the important tool that can be used for designing the optimal routes. This well known combinatorial optimization problem consists of a customer node with deterministic demands, and a central depot which acts as the base of a fleet of vehicles. The customer's demand must be delivered by exactly one vehicle 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, the total demand of the customers assigned to a route does not exceed vehicle capacity, such that minimizing the overall travel cost, taking into account various operational constraints. VRP was first introduced by [5]. A comprehensive overview of the Vehicle Routing Problem can be found in [4] which discusses problem formulations, solution techniques, important variants and applications. There are many other researchers have been working in this area to discover new methodologies. A comprehensive interesting survey of the Vehicle Routing Problem can be found in [4], [3], [36], [37], and [7]. [6] addressed a thorough review of past and recent developments of VRP.

In some cases, particularly, when a customer's demand exceeds the vehicle capacity it is necessary to visit that customer more than once, it requires only a little more thought to see that even when all customer demands are less than or equal to the vehicle capacity, it may be beneficial to use more than one vehicle to serve a customer. For this situation it is necessarily to have split delivery In terms of VRP this type of problem is called the split delivery vehicle routing problem (SDVRP), where the single-visit assumption is relaxed and each customer may be served by more than one vehicle. In this paper it is assumed that the capacity of each vehicle used is homogeny. A survey paper regarding to SDVRP is presented by [10].

Firstly, the SDVRP as the variant of VRP was introduced by [8]. They also gave some properties and a local heuristic search for solving the problem. Due to the structure of the problem, there are two kind of solution approaches have been proposed ,i.e., exact and heuristic methods.

The first exact method for solving SDVRP was addressed by [9]. They introduced a mathematical formulation based on integer programming and solved through a cutting plane approach.[34] and [28] proposed the problem with time windows and present exact approaches based on column generation and branch-and-bound techniques. A column generation approach was presented by [15]. Columns include route and delivery amount information. They solved the pricing sub-problems by a limited-search-with-bound algorithm. Feasible solutions are obtained iteratively by fixing one route once. [21] used branch and price method for the SDVRP formulated as a mixed integer program based on arc-flow consideration. A branch-price-and-cut algorithm was proposed by [24] for solving commodity constrained of SDVRP. They formulated the problem through a set partitioning pattern. [23] present two exact branch-and-cut solution methodologies.

[32] develop a column generation technique to address the problem of scheduling helicopter flights to exchange crews on off-shore platforms. They model the problem as an SDVRP and propose an integer linear programming formulation in which all feasible flight schedules are enumerated in advance, and solve its linear relaxation by means of column generation. [13] proposed a dynamic program with finite state and action spaces, and tackle it by solving the shortest path problem on a digraph whose nodes and arcs correspond to states and transitions between states, respectively. [14] considered the undirected version of the problem and presented new lower bounds together with some polyhedral results for the SDVRP. An integer programming model was introduced and a relaxation of the SDVRP. They showed that all constraints in this relaxation are facet-defining for the convex hull of the incidence vectors of the SDVRP solutions.

[8] developed a heuristic method for SDVRP, which involved a two-stage local search algorithm. Their method was based on the VRP route improvement procedure of [29] The first stage is to find the solution of VRP problem, and then an SDVRP solution is constructed and improved in the second stage. Other heuristic approaches were developed using hybrid methods and metaheuristics. [11] and [20] proposed a tabu search algorithm, a memetic algorithm with population management by [26], three hybrid algorithms due to [16], [12], and [30], a metaheuristic based on the scatter search methodology by [31], an adaptive memory search-

based metaheuristic by [18], a population-based tabu search with vocabulary building approach by [17], a construction heuristic due to [33], a randomized granular tabu search technique by [25]. [22] used the combination of local search and ant colony optimization. A variable neighborhood descent metaheuristic was due to [35]. For a more detailed discussion on heuristics can be found in [19].

This paper concerns with a comprehensive model for the SDVRP incorporated with time windows, fleet and driver scheduling (SDVRPFDTWP) . The basic framework of the vehicle routing part can be viewed as a Heterogeneous split delivery Vehicle Routing Problem with Time Windows (HSDVRPTW) 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 neighborhood heuristic search based on active set constraint is addressed to get the integer feasible solution after solving the continuous model of the problem.

II. MATHEMATICAL FORMULATION

Using graph, SDVRP can be defined as follows. Let G = (V, E) be a complete graph, where V = {v_0, v_1, …, v_n} is the vertex set and E = {(v_i, v_j): v_i, $v_j \in$V, $i \neq j$, $j \neq 0$} is the set of traversing route. Vertex v_0 represents the central depot where a fleet of vehicles is located. The vehicles are assumed to be identical with maximum carrying load equal to Q. The remaining n vertices of V \ {v_0} represent the customer set. Each customer vertex is associated a non-negative known demand q_i, whereas with each arc (v_i, v_j) ∈ E is associated a cost c_{ij} which corresponds to the cost (travel time, distance) for traversing from v_i to v_j. Also, it is assumed that the triangle inequality holds. The delivery services for each vehicle must start and end its traversing route at the depot, v_0.

The demand of a customer must be satisfied and may be fulfilled by more than one vehicle. In order to meet a customer's demand split delivery is needed. One of the reason this condition occur is that customer's demand exceeds vehicle capacity. Mathematically, it can be written as $Q < q_i$.

The objective of the problem is to design the set of Hamiltonian paths to serve all customers such that: the number of vehicles used is minimized, as well as to minimize the total distance or 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 $q_i^t = p_i^t = d_i^t = 0$. The travel time between customer *i* and *j* is given by c_{ij} . Denote the cost coefficients of the travel time of the drivers by *A*.

We define binary variable x_{ijk}^t to be 1 if vehicle *k* travels from node *i* to *j* 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.

This notations used are given as follows :

Set:

T	The set of workdays in the planning horizon,
D	The set of drivers,
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} ∪ *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 $t \in T_i$,
$[g_l^t, h_l^t]$	The start time and the latest ending time of driver $l \in D$ on day $t \in T$,

β_i^t Delivery quantity for customer i on day $t \in T_i$,

H The maximum working duration for each driver over the planning horizon,

F The maximum elapsed driving time without break,

G The duration of the break for drivers,

A_1, A_2, A_3 Costs

Variables:

Binary

x_{ilk}^t Equal to 1 if vehicle $k \in K$ travels from node $i \in N_0$ to $j \in N_0$ with driver $l \in D$ on day $t \in T$,

w_k^t Equal to 1 if vehicle $k \in K$ delivers order,

z_{ik}^t Equal to 1 if vehicle $k \in K$ takes break after serving node $i \in N_0$ on day $t \in T$,

y_{lk}^t Equal to 1 if vehicle $k \in K$ is assigned to driver $l \in D$ on day $t \in T$

Continuous

v_{ik}^t The time at which vehicle $k \in K$ starts service at 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 $t \in T$,

r_l^t The total working duration of driver $l \in D$ on day $t \in T$.

III. THE MODEL

The mathematical formulation for this problem can be formulated as a mixed integer programming model.

$$Min\ A_1.\sum_{i\in N_0}\sum_{l\in D}\sum_{t\in T}\sum_{k\in K} x_{ilk}^t + A_2.\sum_{k\in K}\sum_{t\in T} w_k^t + A_3.\sum_{l\in D}\sum_{k\in K}\sum_{t\in T} y_{lk}^t \qquad (1)$$

Subject to :

$$\sum_{j\in N} x_{ojk}^t = 1 \qquad \forall\, k \in K, t \in T \qquad (2)$$

$$\sum_{k\in K}\sum_{j\in N_0} x_{ijk}^t \geq 1 \qquad \forall\, i \in N, t \in T_i \qquad (3)$$

$$\sum_{k\in K} x_{ijk}^t - \sum_{k\in K} x_{jik}^t = 0 \quad \forall i, j \in N_0, t \in T \qquad (4)$$

$$\sum_{k\in K} x_{ijk}^t = \sum_{k\in K} w_k^t \qquad \forall i, j \in N_0, t \in T \qquad (5)$$

$$\sum_{k\in K} w_k^t \leq 1 \qquad \forall\, t \in T \qquad (6)$$

$$\sum_{i\in N}\sum_{j\in N_0} q_i^t x_{ijk}^t \leq Q_k \qquad \forall\, k \in K, t \in T \qquad (7)$$

$$\sum_{i\in N}\sum_{j\in N_0} p_i^t x_{ijk}^t \leq P_k \qquad \forall\, k \in K, t \in T \qquad (8)$$

$$u_{jk}^t \geq u_{ik}^t + c_{ij} - M(1-x_{ijk}^t) - Mz_{ik}^t \qquad \forall i, j \in N_0,\ k \in K, t \in T \qquad (9)$$

$$u_{jk}^t \geq c_{ij} - M(1-x_{ijk}^t) \quad \forall i, j \in N, k \in K, t \in T \qquad (10)$$

$$u_{ik}^t + \sum_{j\in N_0} c_{ij} x_{ijk}^t - F \leq Mz_{ik}^t \quad \forall i \in N_0,\ k \in K, t \in T \qquad (11)$$

$$v_{jk}^t \geq v_{ik}^t + d_i^t + c_{ij} + G \cdot z_{ik}^t - M(1-x_{ijk}^t) \qquad \forall i, j \in N_0,\ k \in K, t \in T \qquad (12)$$

$$b_i \geq v_{ik}^t \geq a_i \qquad \forall i \in N,\ k \in K, t \in T_i \qquad (13)$$

$$v_{0k}^t \geq \sum_{l\in D}(g_l^t \cdot y_{lk}^t) \qquad \forall k \in K, t \in T \qquad (14)$$

$$v_{n+1,k}^t \leq \sum_{l\in D}(h_l^t \cdot y_{ik}^t) \qquad \forall k \in K, t \in T \qquad (15)$$

$$r_l^t \geq v_{n+1,k}^t - g_l^t - M(1-y_{lk}^t) \qquad \forall\, l \in D, k \in K, t \in T_i \qquad (16)$$

$$\sum_{t\in T_l} r_l^t \leq H \qquad \forall\, l \in D \qquad (17)$$

$$x_{ilk}^t, w_i^t, z_{ik}^t, y_{lk}^t \in \{0,1\} \qquad \forall i, j \in N_0, l \in D,\ k \in K, t \in T \qquad (18)$$

$$v_{ik}^t, u_{ik}^t, r_l^t \geq 0 \quad \forall i, j \in N_0, l \in D,\ k \in K, t \in T \qquad (19)$$

The objective function (1) minimizes the total cost incurred over the planning horizon.

Constraints (2) to ensure that each customer must be visited by one vehicle on each of its delivery days. Constraints (3) impose that each customer node is visited at least once. Flow conservation of vehicles is presented in Constraints (4). The variables x and w need to be linked as shown in Constraints (5). While Constraints (6) is to make sure that every vehicle delivers at least one order per customer. Constraints (7-8) guarantee that the vehicle capacities are respected in bothe weight and volume.

Constraints (9-10) define the elapsed driving time as a function of binary variable z_{ik}^t Meaning that, for the vehicle *k* travelling from customer node *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., $u_{jk}^t \geq u_{ik}^t + c_{ij}$) if the vehicle does not take a break at customer *i* (i.e., $z_{ik}^t = 0$). Otherwise, if the vehicle takes a break at customer *i* (i.e., $z_{ik}^t = 1$), the elapsed driving time at *j* will be constrained by (10) which make sure it is greater than or equal to the travel time between *i* and *j* (i.e., $u_{jk}^t \geq c_{ij}$). Constraints (11) guarantee that the elapsed driving time never exceeds an upper limit *F* by imposing a break at customer *i* (i.e., $z_{ik}^t = 1$) if driving from customer *i* to its successor results in a elapsed driving time

Constraints (12) determine the time to start the service at each customer. If node *j* is visited immediately after node *i*, the time v_{jk}^t to start the service at *j* should be greater than or equal to the service starting time v_{ik}^{t} at i plus its service duration d_i^t, the extra service time $e \cdot p_i^t$ if i is visited by an inappropriate vehicle (i.e., w_j^t = 1), the travel time between the two customers c_{ij}, and the break time G if the driver takes a break after serving I (i.e., z_{ik}^{t} = 1). Constraints (13) make sure the services start within the customers' time window.

Constraints (14-15) 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 (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 drivers work for no more than a maximum weekly working duration. Constraints (18-19) define the binary and nonnegative variables used in this formulation.

IV. NEIGHBORHOOD SEARCH

An optimal solution of a linear programming problem can be detected using the reduced gradient vector. However, generally, in integer programming the reduced gradient vector, 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.

[1] 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 neighborhood search is conducted through proposed feasible points such that the integer feasible solution would be at the least distance from the optimal continuous solution.

V. THE ALGORITHM

This is a combination of exact and heuristic method. The exact method is used for solving the relaxed problem of the model. The heuristic method, called neighborhood search, can described as follows.

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

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

This Step is to find the direction vector.

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* = {∅} go to Stage 2, otherwise Repeat from step 1.

Stage 2. Pass1 : adjust integer infeasible superbasics (if any) 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.

CONCLUSIONS

This paper was intended to develop efficient technique for solving one of the most economic importance problems in optimizing logistic systems. The aim of this paper was to develop a model of split delivery vehicle routing with Time Windows, Fleet and Driver Scheduling Problem This problem has additional constraint which is the limitation in the weight and volume of vehicles. The proposed algorithm employs nearest neighbor heuristic algorithm for solving the model.

References

[1] 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), 1986, pp. 117-134, 1986.

[2] J.-F. Cordeau, G. Laporte, M.W.F. Savelsbergh, and D. Vigo. *Vehivle Routing*. In Barnhart, C. and Laporte, G., editors, *Transportation*, handbooks in Operation Research and Management Science, Pages 367-428. North-Holland, Amsterdam, 2007.

[3] O. Bräysy and M. Gendreau. "Vehicle Routing Problem with Times Windows, Part II: Metaheuristics". *Transportation Science*, 39(1), 2005, pp. 104-118.

[4] P. Toth and D. Vigo. "The Vehicle Routing Problem Society for Industrial and Applied Matheamtics", Philadelpiha, PA, 2002.

[5] G.B. Dantzig and J.H. Ramser, "The truck dispatching problem", *Management Science*, Vol. 6, 1959, pp. 80.

[6] Gendreau, M., Tarantilis, C.D., "Solving large-Scale Vehicle Routing Problems with Time Windows: The State-of-the-Art". CIRRELT-2010-04. Universite´ de Montreal, Canada, 2010.

[7] Toth, P., Vigo, D. (2nd Eds.). "Vehicle routing : Problems, Methods, and Applications, *Society for Industrial and Applied Mathematics & Mathematical Optimization Society*", Philadelphia PA, 2014.

[8] M. Dror and P. Trudeau, "Savings by Split Delivery Routing," *Transportation Science*, Vol. 23, No. 2, 1989, pp. 141-145.

[9] M. Dror, G. Laporte and P. Trudeau, "Vehicle Routing with Split Deliveries," *Discrete Applied Mathematics*, Vol. 50, No. 3, 1994, pp. 239-254.

[10] C. Archetti and M. G. Speranza, "The Split Delivery Ve-hicle Routing Problem: A Survey. The Vehicle Routing Problem: Latest Advances and New Challenges," *Operations Research/Computer Science Interfaces Series*, Vol. 43, Part I, 2008, pp. 103-122.

[11] C. Archetti, M. Savelsbergh and A. Hertz, "A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem," *Transportation Science*, Vol. 40, No. 1, 2006, pp. 64-73.

[12] C. Archetti, M. Savelsbergh and M. G. Speranza, "To Split or Not to Split: That Is the Question," *Transportation Research Part E*, Vol. 44, No. 1, 2008, pp. 114-123.

[13] C. Lee, M. A. Epelman, C. C. White and Y. A. Bozer, "A Shortest Path Approach to the Multiple-Vehicle Routing Problem with Split Pick-Ups," *Transportation Research Part B*, Vol. 40, No. 4, 2006, pp. 265-284.

[14] J. M. Belenguer, M. C. Martinez and E. A. Mota, "Lower Bound for the Split Delivery VRP," *Operations Research*, Vol. 48, No. 5, 2000, pp. 801-810.

[15] M. Jin, K. Liu and R. O. Bowden, "A Two-Stage Algo-rithm with Valid Inequalities for the Split Delivery Vehi-cle Routing Problem," *International Journal of Production Economics*, Vol. 105, No. 1, 2007, pp. 228-242.

[16] S. Chen, B. Golden and E. Wasil, "The Split Delivery Vehicle Routing Problem: Applications, Test Problems, and Computational Results," *Networks*, Vol. 94, No. 4, 2007, pp. 318-329.

[17] R.E. Aleman and R.R. Hill., "A tabu search with vocabulary building approach for the vehicle routing problem with split demands". *International Journal of Metaheuristics*, 1(1), 2010, pp. 55-80.

[18] R.E. Aleman, X. Zhang, and R.R. Hill., "An adaptive memory algorithm for the split delivery vehicle routing problem". *Journal of Heuristics*, 16(3), 2010, pp. 441-473.

[19] C. Archetti and M.G. Speranza., "Vehicle routing problems with split deliveries". *International Transactions in Operational Research*, 19(1-2), 2012, pp. 3-22.

[20] C. Archetti, M.G. Speranza, and A. Hertz., "A tabu search algorithm for the split delivery vehicle routing problem". *Transportation Science*, 40(1), 2006, pp. 64-73.

[21] M. Salani and I. Vacca C. , "Branch and price for the vehicle routing problem with discrete split deliveries and time windows". *Report TRANSP-OR* 091224, Ecole Polytechnique, 2009.

[22] M. E. McNabb, J. D. Weir, R. R. Hill, and S. N. Hall., "Testing local search move operators on the vehicle routing problem with split deliveries and time windows", *Computers & Operations Research* 56, 2015, 93-109.

[23] C. Archetti, N. Bianchessi, and M.G. Speranza. "Branch-and-cut algorithms for the split delivery vehicle routing problem". *European Journal of Operational Research*, 2014.

[24] C. Archetti, N. Bianchessi, and M.G. Speranza., "A branch-price-and-cut algorithm for the commodity constrained split delivery routing problem", *Computers & Operations Research*, 54, 2015, pp. 1-10.

[25] L. Berbotto, S. Garcia, and F.J. Nogales., "A randomized granular tabu search heuristic for the split delivery vehicle routing problem". *Annals of Operations Research*, 2013, pp. 1-21.

[26] M. Boudia, C. Prins, and M. Reghioui., "An effective memetic algorithm with population management for the split delivery vehicle routing problem". In *Hybrid Metaheuristics*, pages 16-30. Springer, 2007.

[27] A. Ceselli, G. Righini, and M. Salani., "A column generation algorithm for a rich vehicle-routing problem". *Transportation Science*, 43(1), 2009, pp. 56-69.

[28] G. Desaulniers., "Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows". *Operations Research*, 58(1), 2010, pp. 179-192.

[29] M. Dror and L. Levy., "A vehicle routing improvement algorithm comparison of a greedy and a matching implementation for inventory routing". *Computers & Operations Research*, 13(1), 1986, pp. 33-45.

[30] M. Jin, K. Liu, and B. Eksioglu., "A column generation approach for the split delivery vehicle routing problem". Operations Research Letters, 36(2), 2008, pp. 265-270.

[31] E. Mota, V. Campos, and _A. Corberan., "A new metaheuristic for the vehicle routing problem with split demands". In Evolutionary Computation in Combinatorial Optimization, pages 121-129. Springer, 2007.

[32] G. Sierksma and G.A. Tijssen., "Routing helicopters for crew exchanges on off-shore locations". *Annals of Operations Research*, 76, 1998, pp. 261-286.

[33] J.H. Wilck-IV and T.M. Cavalier., "A construction heuristic for the split delivery vehicle routing problem". *American Journal of Operations Research*, 2, 2012, pp. 153.

[34] Gendreau, M., P. Dejax, D. Feillet and C. Gueguen, "Vehicle routing with time windows and split deliveries, Technical Report, Technical report 2006-851, Laboratoire d'Informatique d'Avignon, 2006.

[35] A. F. Han, and Y. Chu., "A multi-start heuristic approach for the split delivery vehicle routing problem with minimum delivery amounts", *Transportation Research Part E* 88, 2016, pp. 11-31.

[36] Vidal, T., T. G. Crainic, M. Gendreau, C. Prins. "Heuristics for multi-attribute vehicle routing problems: A survey and synthesis." CIRRELT-2012-05.

[37] Kumar, S. N., R. Pannerselvam, "A survey on the vehicle routing problem and its variants", *Intelligent Information Management*, 4, 2012,pp. 66-74..