International Conference
on Industrial Engineering and Systems Management
IESM 2011
May 25 - May 27
METZ - FRANCE

Two objective functions for a real life Split Delivery Vehicle Routing Problem ⋆

Marc Uldry [a], Marino Widmer [a], Alain Hertz [b]

[a] *Decision Support & Operations Research, Department of Informatics, University of Fribourg*

[b] *Département de mathématiques et de génie industriel, École Polytechnique de Montréal*

Abstract

Different products have to be delivered by trucks. Due to the trucks capacity, each order is split into one or more deliveries supplied from a main depot or from local depots. Simple and complex constraints have to be satisfied as drivers and trucks availability or the fact that only specific trucks can be loaded in local depots. The problem is formulated as a MIP and considers two different objective functions. The first one defines a vehicle routing problem where the total travel time is minimized while the second one defines a VRP where the number of different trucks supplying each individual customer is minimized.

Key words: Vehicle Routing Problem, Split Deliveries, Mixed Integer Programming

1 Introduction

In the classical Vehicle Routing Problem (VRP), a fleet of vehicles has to supply a set of customers from a depot with the objective of minimizing the total traveled distance. Each customer has to be visited exactly once and the total demand of customers visited by a vehicle must not exceed its capacity. An overview of resolution methods can be found in [12] and [5]. Extensions of this problem, named Split Delivery Vehicle Routing Problems (SDVRP), are studied for over 20 years and were first introduced in [8]. In this seminal paper, Dror and Trudeau split deliveries such that each customer can be visited by more than one vehicle. Due to the complexity of the problem (it is NP-hard), a two-stage algorithm including five interconnected subroutines is developed, but it does not guarantee an optimal solution. The relaxation results in savings in the total traveled distance and in the number of vehicles used in the solution. An empirical study is presented in [1]. A similar heuristic is implemented in [13] to solve a feed distribution problem formulated as a collection of split delivery capacitated rural postman problem with time windows on arcs. In [7], the Split Delivery Vehicle Routing Problem is formulated as an integer linear program including constraints on subtours elimination and is decomposed in subproblems solved by simplex and branch and bound methods. A mathematical formulation and a heuristic are presented in [10] which include grid network distances and time window constraints. A lower bound based on a polyhedral study of the problem is proposed in [3] and a branch and price approach is presented in [9]. In [11] and [2], Tabu Search metaheuristics using a memory-based search strategy which give quickly good

⋆ This paper was not presented at any other revue.
Corresponding author M. Uldry. Tel. +41 26 300 93 53. Fax +41 26 300 97 26.
Email addresses: marc.uldry@unifr.ch (Marc Uldry), marino.widmer@unifr.ch (Marino Widmer), alain.hertz@gerad.ch (Alain Hertz).

results are implemented. While in their two-stage algorithm Ho and Haugland improve a VRP solution by introducing and eliminating splits, Archetti et al. improve an initial feasible solution by removing and inserting a customer at each iteration of their method. Applications and resolution methods of the SDVRP are reviewed in [4] before the presentation of a heuristic that combines a mixed integer program and a record-to-record travel algorithm. In [6], an exact branch-and-price-and-cut method is proposed, which is a branch-and-bound algorithm where the lower bounds are computed by a column generation algorithm and the linear relaxations are tightened by adding cutting planes.

This article considers a real life problem, which can be described as a special Split Delivery Vehicle Routing Problem where the demand of each customer is mostly greater than the capacity of any available vehicle. This explains why each demand needs to be split into one or more deliveries. The fleet is not homogeneous and the vehicles have therefore various capacities. There is also a main depot where the vehicles can load products and have to start and end their journey, and local depots where only some special vehicles with a specific equipment can load products. A vehicle can only supply one customer between two loads. A similar problem is treated in [14] but several main differences with our problem may be highlighted. Indeed, in her PhD thesis, Schmid considers the following assumptions: delivered products are perishable goods; some vehicles with specialized unloading equipment have to assist other vehicles during their unloading operation; each vehicle is assigned to a specific depot where it has to start and end its journey but can visit the other depots during the day. Formulated as a mixed integer program, two hybrid approaches are developed to solve instances of the problem, one with local branching and variable neighborhood search heuristics, one based on a multi-commodity network flow model.

This paper is organized as follows. Section 2 gives a detailed presentation of the problem placed in its context. Section 3 provides a formulation of the problem including three mixed integer programs. Section 4 presents and discusses some numerical results of the implemented algorithm. The conclusion follows in Section 5.

2 Problem Description

A factory produces different kinds of products. Every day, products have to be delivered by trucks to several customers. Due to the trucks capacity, each order is split into one or more deliveries. These deliveries are supplied either from a main depot, the factory, where quantities of products are unlimited or, for special trucks with specific equipment, from local depots located in railways stations where quantities of products are limited. Each delivery has a loading time at a depot, a travel time between this depot and the concerned customer, an unloading time at this customer and a back travel time between this customer and a depot. The loading and unloading times depend on the used truck, on the kind of product and on the equipment of the location. The travel and back travel times depend on the used truck and on the type of road to the locations. Each truck starts and ends its daily activity at the factory such that the first delivery of a truck is loaded at the factory and after its last delivery, each truck has to drive back to the factory.

Due to an heterogeneous fleet of vehicles, each truck has its own capacity and can only deliver some customers. To spare time, the trucks, which have also a daily availability, can be "preloaded" the previous day with a kind of product that will be probably ordered by a customer. Note that the scheduler can impose a first delivery and fix various tasks during the scheduling to some trucks like maintenance operations, for example. Although most of the vehicles belongs to the enterprise, some trucks can be rented to a third party if necessary which involves naturally additional costs. Each truck-driver belonging to the enterprise can only drive some trucks, supply some customers and handle some kinds of product. Every day, each available truck is assigned to an available truck-driver by the scheduler.

Although the *natural* objective of this problem is to determine a set of routes that minimizes the total travel duration, the enterprise prefers that the number of trucks supplying a same customer is minimized.

3 Problem Formulation

To solve this split delivery vehicle routing problem, the idea consists in decomposing the problem into three sub-problems. First, each order is allocated to the most appropriate depot. Then, deliveries are created and allocated to the trucks. Finally, the deliveries are sequenced for each truck.

3.1 Allocation of each order to a specific depot

A set T of product types and a set J of depots including the factory and the railways stations are considered. The stock of product type t available in depot j before the allocation of the deliveries to the trucks is denoted by S_{jt}. Although the stock of each product type is unlimited in reality at the factory, it is limited to 10^{18} in the MIP models. A set I of orders is also considered. Q_{it} denotes the quantity of product type t required by order i. D_{ij} corresponds to the travel time between the customer of order i and depot j, i.e. the duration of a delivery including a loading time, a travel time from j to i, an unloading time and a back travel time from i to j.

The following set of Boolean variables is considered in the MIP models:

$$x_{ij} = \begin{cases} 1 \text{ if order } i \text{ is allocated to depot } j \\ 0 \text{ otherwise} \end{cases}$$

Thanks to the elements defined previously, this first sub-problem can be formulated as a mixed integer linear program:

The objective (1) is to minimize the total travel time between the customers and their allocated depot:

$$\min z(x) = \sum_{i \in I} \sum_{j \in J} D_{ij} x_{ij} \tag{1}$$

Constraints (2) impose that each order is associated to only one depot:

$$\sum_{j \in J} x_{ij} = 1 \qquad \forall i \in I \tag{2}$$

Constraints (3) ensure that the total demand of orders assigned to a depot in product types does not exceed the depot's capacity:

$$\sum_{i \in I} (Q_{it} x_{ij}) \leq S_{jt} \qquad \forall j \in J, \forall t \in T \tag{3}$$

3.2 Allocation of the deliveries to the trucks

A set J of depots including the factory and the railways stations is considered. To deliver the different kinds of products from the depots, a set K of trucks is available but only trucks with specific equipment denoted by a subset $KF \subseteq K$ can be loaded in the railways stations. The capacity and the daily availability of truck k are respectively denoted by q_k and e_k. A time margin is required in order to satisfy the constraint that each special truck $k \in KF$ has to start and end its tour at the factory. A weighting factor $\theta = 0.8$ is therefore considered which allows a margin of 20% on the truck availability. A factor O_k penalizes the use of any truck k belonging to a third party. In the MIP models, $O_k = 1$ for owned trucks and $O_k = 10$ for rented trucks.

As a customer can have several orders, a set H of customers and a set I of orders are considered. Although most of orders can be more or less satisfied (a maximal remainder $\phi_{\min}$ of non delivered products or a maximal surplus $\phi_{\max}$ of delivered products can be accepted), some orders denoted by a subset $IE \subseteq I$ need to be exactly satisfied in only one delivery. B_{hi} indicates if customer h is associated with order i and d_i denotes the quantity required by order i. c_{ik} corresponds to the travel time between the customer of order i and his allocated depot with truck k, i.e. the duration of a delivery including a loading time, a travel time from the allocated depot of i to i, an unloading time and a back travel time from i to its allocated depot. N_{ij} denotes if order i is associated to depot j (defined in Subsection 3.1). Taking into account the specifications of the trucks and the skills and availability of their driver, M_{ik} indicates if order i can be delivered by truck k. To simplify the resolution of the problem instances, an upper bound l_{ik} corresponding to $\lceil d_i/q_k \rceil$, the maximum number of deliveries needed to satisfy order i with the unique use of truck k, is computed.

To spare time on the loading time of the first deliveries, a few trucks are sometimes preloaded with a product type before the allocation of the deliveries to the trucks. A subset $KP \subseteq K$ of preloaded trucks is therefore considered. P_{ik} indicates if a delivery of order i can be allocated to preloaded truck k, i.e. if the product type of order i is the same as the preloaded product type of truck k. It happens that preloaded trucks are already associated with orders. A_{ik} indicates therefore if at least a delivery of order i is allocated or not to truck k. Additional restrictions related to the number of trucks or depots to use can be considered in the models. Indeed, Nr denotes the maximum number of different trucks to use for each customer and Nd, the maximum number of different depots to use for each truck. All available trucks are not necessary useful to do the deliveries. To compact these ones, the maximum number of different trucks to use Nk is therefore considered.

The six following sets of integer and Boolean variables are considered in the MIP models:

$$u_{ik} = \text{the number of deliveries done by truck } k \text{ for order } i \text{ from its allocated depot}$$

$$x_{hk} = \begin{cases} 1 \text{ if at least a delivery is done for customer } h \text{ by truck } k \\ 0 \text{ otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 \text{ if at least a delivery is done for order } i \text{ by truck } k \\ 0 \text{ otherwise} \end{cases}$$

$$z_{jk} = \begin{cases} 1 \text{ if truck } k \text{ supplies an order from depot } j \\ 0 \text{ otherwise} \end{cases}$$

$$v_{ik} = \begin{cases} 1 \text{ if order } i \text{ is allocated to preloaded truck } k \\ 0 \text{ otherwise} \end{cases}$$

$$w_{k} = \begin{cases} 1 \text{ if truck } k \text{ is used} \\ 0 \text{ otherwise} \end{cases}$$

Thanks to the elements defined previously, this second sub-problem can be formulated as two relatively similar mixed integer linear programs. Presented in the two following subsections, each one considers a different objective.

3.2.1 First objective

The first objective (4) is to minimize the total travel time of the deliveries taking into account the penalty factor O_k related to each truck for its use:

$$\min z(u) = \sum_{i \in I} \sum_{k \in K} O_k c_{ik} u_{ik} \tag{4}$$

Constraints (5) and (6) impose that the demand of order $i \in IE$ is exactly or more satisfied in only one delivery while contraints (7) accept for order $i \in I \setminus IE$ multiple deliveries and a maximal remainder $\phi_{\min}$ of non delivered product:

$$\sum_{k \in K} q_k u_{ik} \geq d_i \qquad \forall i \in IE \tag{5}$$

$$\sum_{k \in K} u_{ik} = 1 \qquad \forall i \in IE \tag{6}$$

$$\sum_{k \in K} q_k u_{ik} \geq d_i - \phi_{\min} \qquad \forall i \in I \setminus IE \tag{7}$$

Constraints (8) and (9) ensure that the availability of each truck is respected. A weighting factor $\theta = 0.8$ is imposed for each special truck $k \in KF$ accessing the railways stations to keep an availability margin of 20% to start and end its tour at the factory:

$$\sum_{i \in I} c_{ik} u_{ik} \leq \theta e_k \qquad \forall k \in KF \tag{8}$$

$$\sum_{i \in I} c_{ik} u_{ik} \leq e_k \qquad \forall k \in K \setminus KF \tag{9}$$

Constraints (10) state that an order can be delivered by a truck only if the truck and its driver are able to do it:

$$y_{ik} \leq M_{ik} \qquad \forall i \in I, \forall k \in K \tag{10}$$

Constraints (11) and (12) impose that the number of deliveries done by a truck for an order is feasible, strictly positive and lower than l_{ik}, the maximum number of deliveries needed with the unique use of the truck:

$$u_{ik} \geq y_{ik} \qquad \forall i \in I, \forall k \in K \tag{11}$$

$$u_{ik} \leq l_{ik} y_{ik} \qquad \forall i \in I, \forall k \in K \tag{12}$$

Constraints (13) state if a delivery of order i can be associated to a preloaded truck $k \in KP$ while constraints (14) ensure that at least one preloaded truck $k \in KP$ is associated to a delivery of an order:

$$v_{ik} = P_{ik} y_{ik} \qquad \forall i \in I, \forall k \in KP \tag{13}$$

$$\sum_{i \in I} v_{ik} \geq 1 \qquad \forall k \in KP \tag{14}$$

Constraints (15) impose that at least one delivery of order i is supplied with truck k if it is beforehand defined:

$$u_{ik} \geq A_{ik} \qquad \forall i \in I, \forall k \in K \tag{15}$$

Constraints (16) state the relations between the depots and the trucks while constraints (17) impose the maximal number of different depots Nd that a special truck $k \in KF$ can access during its tour:

$$z_{jk} \geq N_{ij} y_{ik} \qquad \forall i \in I, \forall j \in J, \forall k \in K \tag{16}$$

$$\sum_{j \in J} z_{jk} \leq Nd \qquad \forall k \in KF \tag{17}$$

Constraints (18) state that a truck is used only if it does at least a delivery and constraint (19) imposes the maximal number of different trucks Nk to use do the allocation of deliveries:

$$w_k \geq y_{ik} \qquad \forall i \in I, \forall k \in K \tag{18}$$

$$\sum_{k \in K} w_k \leq Nk \tag{19}$$

3.2.2 Second objective

Also subject to the constraints (5)–(19) detailed previously, the second objective (20) is to minimize the number of different trucks supplying the customers taking into account the penalty factor O_k related to each truck for its use:

$$\min z(x) = \sum_{h \in H} \sum_{k \in K} O_k x_{hk} \tag{20}$$

Constraints (21) impose that the quantity of delivered product does not exceed the maximal surplus $\phi_{\max}$ for each order:

$$\sum_{k \in K} q_k u_{ik} \leq d_i + \phi_{\max} \qquad \forall i \in I \tag{21}$$

Constraints (22) define if customer h is supplied or not by truck k during the allocations of deliveries:

$$x_{hk} \geq B_{hi} y_{ik} \qquad \forall h \in H, \forall i \in I, \forall k \in K \tag{22}$$

3.3 Sequencing of the deliveries for each truck

A set I of n deliveries allocated to the truck (defined in Subsection 3.2) is considered. If the truck is preloaded, a subset $P \subseteq I$ denotes the deliveries that can be associated to the preload of the truck. If the first delivery of the truck is beforehand defined, a subset $A \subseteq I$ containing only this delivery is considered. The travel time between the customer of delivery i and the depot of delivery j is denoted by c_{ij}.

The two following sets of Boolean and integer variables are considered in the MIP model:

$$s_{ij} = \begin{cases} 1 \text{ if the visit of delivery } i \text{ preceeds the visit of delivery } j \\ 0 \text{ otherwise} \end{cases}$$

$$u_i = \text{the rank of the visit of delivery } i$$

Thanks to the elements defined previously, this third sub-problem can be formulated as a mixed integer linear program:

The objective (23) is to minimize the total travel time between the deliveries:

$$\min z(s) = \sum_{i \in I} \sum_{j \in I} s_{ij} c_{ij} \tag{23}$$

Constraints (24) and (25) impose that each delivery is done only once:

$$\sum_{\substack{i \in I \\ i \neq j}} s_{ij} = 1 \qquad \forall j \in I \tag{24}$$

$$\sum_{\substack{j \in I \\ i \neq j}} s_{ij} = 1 \qquad \forall i \in I \tag{25}$$

Constraints (26) prohibit any subtour:

$$u_i - u_j + n \cdot s_{ij} \leq n - 1 \qquad \forall i \in I, \forall j \in J, i > 1, j > 1 \tag{26}$$

If the truck is preloaded, constraint (27) imposes that only one delivery $p \in P$ is allocated to the preload:

$$\sum_{p \in P} s_{1,p} = 1 \tag{27}$$

If the first delivery of the truck is beforehand defined, constraints (28) impose that the defined delivery $a \in A$ is respected:

$$s_{1,a} = 1 \qquad \forall a \in A \tag{28}$$

The MIP models described in this section are used for the computational experiments presented in next section.

4 Problem Resolution

This section contains some details about the implementation of an algorithm followed by computational experiments to evaluate its performance.

4.1 Algorithm implementation

To solve any instance of the Split Delivery Vehicle Routing Problem presented in this article, an algorithm has been implemented in Delphi Programming Language using CPLEX Interactive Optimizer 12.2.0.0 (CPLEX). The sub-problems detailed in previous section are solved sequentially such that the output of each sub-problem is used as an input of the following sub-problem. For each sub-problem, the algorithm first prepares the structure and data depending on the objective to achieve and the imposed constraints, then solves the created problem instance with help of CPLEX and its mixed integer optimizer, and finally post-optimizes the solution if needed.

Although most parameters used by the algorithm are set manually, this one can estimate, if needed, the maximum number of trucks to use before solving an instance of the second sub-problem. This avoids to allocate the deliveries on too many different vehicles by compacting them on a sufficient number of trucks. The estimation of this parameter is defined in (29) and refers to the notation exposed in Section 3.2 except for ζ, which corresponds to 1 for the first objective, where the total travel time of the deliveries has to be minimized, and to 2 for the second objective, where the number of different trucks supplying the customers has to be minimized.

$$Nk = \min(|K|, \max(|KP|, \left\lceil \frac{\sum d_i}{\frac{\sum q_k}{|K|} \cdot \left\lceil \frac{(\sum e_k)\cdot(\theta|KF| + |K \setminus KF|)}{\frac{\sum\sum c_{ik} d_i}{\sum d_i}\cdot|K|} \right\rceil} \cdot 1.1 \right\rceil + \zeta)) \tag{29}$$

The maximum number of trucks to use cannot exceed the number of trucks available and cannot be lower than the number of preloaded trucks. That is why the value of the estimated parameter must be between these two bounds. To estimate the number of trucks to use, the total quantity ordered is devided by the average capacity of a truck and by the average number of deliveries that can be allocated to a truck. Averages are not always good estimators, and empirical tests have indeed shown that adding 10% to the obtained value and finally adding an extra truck (for the first objective) or two extra trucks (for the second objective) tends to produce better results.

4.2 Computational experiments

Computational experiments to evaluate the performance of the algorithm have been done on a PC Core 2 Duo, CPU 3.00 GHz, 4 GB RAM. Thirty instances of the problem have been built from real life orders spread over ten days. The number of orders and the quantities ordered have been decreased and increased such that three different instances have been created for each day. Each instance has been tested with the two different objective functions defined in Sections 3.2.1 and 3.2.2 during four different time periods, i.e. 900, 1'800, 3'600 and 21'600 seconds. The parameter settings for the algorithm, presented in table 1, indicate that up to three different trucks can be used to supply a same customer (this constraint is only valid for the first objective) and that up to two different local depots can be visited per truck to restrict the travel area of each truck. They also indicate that each order must be more or less satisfied, that is why a maximal remainder of five tons and a maximal surplus of five tons (this constraint is only valid for the second objective) have been accepted for each delivered product, and that the availability of each special truck with specific equipment has been weighted to eighty percent to keep a flexibility of twenty percent for possible extension of the back travel times between deliveries.

Table 1
Parameter settings for the algorithm

Description	Value
Maximum number of different trucks to use for each customer	$Nr = 3$
Maximum number of different depots to use for each truck	$Nd = 2$
Maximal remainder of any order	$\phi_{min} = 5'000$ kg
Maximal surplus of any order	$\phi_{max} = 5'000$ kg
Weighting factor bound to the availability of each special truck	$\theta = 80\%$

The solutions of the computational experiments are detailed in table 2 for the first objective and in table 3 for the second objective. For each problem instance we indicate the number of orders, the number of customers, the number of preloaded trucks, the time used by the CPU to solve the problem instance, the gap in percent between the actual solution and the best solution or the lower bound, the total travel duration, the number of deliveries, the number of trucks used, the number of different trucks supplying the customers and the maximal surplus of delivered product among the orders. The symbols '-' and '*' respectively indicate that there is no feasible solution and that there is not enough memory to solve the problem instance.

For the first sub-problem (allocation of each order to a specific depot) and the third sub-problem (sequencing of the deliveries for each truck), an optimal solution of each mixed integer linear program has been found in a few seconds. But this is not the case for the second sub-problem (allocation of the deliveries to the trucks). Among the 30 problem instances, the algorithm has found for the first objective 26 feasible solutions after 900, 1'800, 3'600 and 21'600 seconds. Among these feasible solutions, the algorithm has found 9 optimal solutions in less than 60 seconds, 9 optimal solutions between 61 and 120

Table 2
Solutions of computational experiments (first objective : min duration)

Problem	$\lvert I\rvert$	$\lvert H\rvert$	$\lvert KP\rvert$	CPU	Gap	Duration	Deliveries	$\lvert K_{\text{U}}\rvert$	$\lvert Kd_{\text{U}}\rvert$	Surplus
1	16	16	0	52	0.00%	3'340	25	8	19	18'000
2	24	21	0	66	0.00%	5'215	35	11	30	18'000
3	31	21	0	102	0.00%	7'670	53	16	42	18'500
4	18	15	1	53	0.00%	3'775	27	8	22	15'200
5	29	21	1	81	0.00%	6'660	45	14	39	14'700
6	35	21	1	*439	1.01%	8'355	58	18	38	15'200
7	18	16	0	57	0.00%	3'305	25	8	21	19'000
8	31	25	0	*4'087	0.79%	6'600	47	14	38	16'200
9	37	25	0	*1'035	1.31%	8'380	59	17	45	14'200
10	14	13	0	51	0.00%	3'525	26	9	20	13'600
11	25	22	0	72	0.00%	6'665	47	15	41	11'200
12	31	22	0	599	0.00%	8'485	60	18	42	6'960
13	16	14	3	52	0.00%	3'785	28	9	22	16'680
14	30	27	3	*7'399	-	-	-	-	-	-
15	36	27	3	*619	-	-	-	-	-	-
16	18	16	2	58	0.00%	4'065	28	9	21	19'400
17	34	28	2	90	0.00%	7'315	50	15	41	9'500
18	39	28	2	*1'092	0.95%	9'095	61	19	54	9'000
19	17	15	1	57	0.00%	3'680	26	9	17	8'500
20	28	25	1	78	0.00%	6'800	45	14	39	10'060
21	36	25	1	952	0.00%	9'260	62	18	50	18'500
22	16	15	5	51	0.00%	3'320	23	8	20	16'830
23	27	24	5	75	0.00%	6'095	41	13	32	9'000
24	34	24	5	416	0.00%	8'255	56	17	45	9'500
25	15	14	3	54	0.00%	2'870	22	7	17	9'000
26	30	26	3	*415	-	-	-	-	-	-
27	36	26	3	*595	-	-	-	-	-	-
28	14	13	5	67	0.00%	4'940	38	12	29	8'650
29	21	19	5	89	0.00%	6'640	49	15	35	11'780
30	28	19	5	*1'490	2.29%	9'175	67	19	56	4'700

seconds, 2 optimal solutions between 121 and 600 seconds and an optimal solution in 952 seconds, i.e. a total of 21 optimal solutions (70.0% of the instances). Furthermore, 7 optimal solutions found before 90 seconds concern real life orders while the other optimal solutions concern theoretical case studies where the number of orders and the quantities ordered have been decreased or increased compared to real life orders. No feasible solution has been found for 4 problem instances (13.3% of the instances) despite six hours of execution of the algorithm. Among these problem instances, two concern real life orders while the others concern theoretical case studies where the number of orders and the quantities ordered have been increased compared to real life orders. For the second objective, the algorithm has found 28 feasible solutions after 900, 1'800, 3'600 and 21'600 seconds. Among these feasible solutions, the algorithm has found 10 optimal solutions in less than 60 seconds, 9 optimal solutions between 61 and 120 seconds, 3 optimal solutions between 121 and 600 seconds, an optimal solution in 1'301 seconds and an optimal solution in 4'451 seconds, i.e. a total of 24 optimal solutions (80.0% of the instances). Furthermore, 10 optimal solutions found before 1'301 seconds concern real life orders while the other optimal solutions concern theoretical case studies where the number of orders and the quantities ordered have been decreased or increased compared to real life orders. No feasible solution has been found for 2 problem instances (6.7% of the instances) despite six hours of execution of the algorithm. These problem instances only concern theoretical case studies where the number of orders and the quantities ordered

Table 3
Solutions of computational experiments (second objective : min trucks)

Problem	$\|I\|$	$\|H\|$	$\|KP\|$	CPU	Gap	Duration	Deliveries	$\|K_{\text{U}}\|$	$\|Kd_{\text{U}}\|$	Surplus
1	16	16	0	49	0.00%	3'665	26	9	17	4'440
2	24	21	0	67	0.00%	5'550	36	12	24	4'440
3	31	21	0	0	-	-	-	-	-	-
4	18	15	1	52	0.00%	4'075	28	10	15	4'950
5	29	21	1	99	0.00%	6'830	46	15	25	4'380
6	35	21	1	133	0.00%	8'420	59	18	26	4'950
7	18	16	0	55	0.00%	3'810	25	9	16	4'300
8	31	25	0	80	0.00%	6'990	48	15	28	4'400
9	37	25	0	*592	7.59%	8'630	60	19	31	4'700
10	14	13	0	51	0.00%	3'705	27	8	14	4'800
11	25	22	0	72	0.00%	7'220	49	16	26	4'800
12	31	22	0	0	-	-	-	-	-	-
13	16	14	3	51	0.00%	4'060	29	10	15	4'680
14	30	27	3	1'301	0.00%	8'610	58	18	34	4'300
15	36	27	3	*967	35.06%	10'610	70	21	43	3'580
16	18	16	2	54	0.00%	4'670	29	10	17	4'500
17	34	28	2	90	0.00%	7'505	50	16	30	4'500
18	39	28	2	4'451	0.00%	9'535	64	19	33	4'380
19	17	15	1	57	0.00%	3'590	26	10	15	4'300
20	28	25	1	77	0.00%	7'055	47	15	27	4'650
21	36	25	1	313	0.00%	9'545	63	19	35	4'560
22	16	15	5	51	0.00%	3'555	25	9	16	3'800
23	27	24	5	74	0.00%	6'390	42	14	26	4'500
24	34	24	5	326	0.00%	8'650	58	18	30	4'380
25	15	14	3	53	0.00%	3'295	23	9	15	4'440
26	30	26	3	86	0.00%	8'200	51	17	31	4'500
27	36	26	3	*1'073	25.38%	9'905	66	20	34	4'500
28	14	13	5	58	0.00%	5'365	39	13	17	4'600
29	21	19	5	74	0.00%	7'045	51	15	23	5'000
30	28	19	5	*1'217	12.39%	9'170	69	19	29	4'330

have been increased compared to real life orders.

The total travel duration is nearly always lower for the first objective than for the second objective. Exceptions can however be found for the 19th problem instance after an algorithm execution of 900, 1'800, 3'600 and 21'600 seconds and for the 30th problem instance after an algorithm execution of 1'800, 3'600 and 21'600 seconds. This can be explained by the back travel times between the deliveries which are unknown when solving the second sub-problem and only known when solving the third sub-problem. The number of different trucks used is always lower for the second objective than for the first objective. For each problem instance, the number of deliveries is often identical for the two different objectives. There is only sometimes a maximal difference of 3 deliveries. Furthermore, we can observe that one more truck is mostly used for the second objective compared to the first objective because of the value of the estimator related to the maximum number of trucks to use which differs according to the objective to achieve. The use of the number of allowed trucks gives indeed more flexibility to achieve the second objective. Concerning the maximal surplus of delivered product among the orders, the value according to the first objective is always greater than the value according to the second objective which is bounded to be lower or equal than 5'000 kilos. The delivered quantity corresponds therefore less to the ordered quantity for the first objective than for the second objective.

5 Conclusion

In this paper, two objective functions for a real life Split Delivery Vehicle Routing Problem have been presented. In order to be solved, this problem has been decomposed into three sub-problems formulated with the help of mixed integer linear programming. An algorithm using CPLEX has then been implemented to solve sequentially each sub-problem. Computational experiments have shown that optimal solutions are always found for the first and third sub-problem. For the second sub-problem, optimal solutions have often been found whatever the objective to achieve. Most of them concern small or normal problem instances which reflect reality. Good feasible solutions could be obtained in most other cases, the model with the first objective giving typically better solutions than the second one. Two of the six unfeasible solutions concern normal problem instances while four of them concern big problem instances. Regarding the solution improvement between an algorithm execution of 900 and 21'600 seconds, there is little change for all of the problem instances.

Although this is not take into account in the instances used for the computational experiments, the use of each truck can be penalized according to its emissions of carbon dioxide (CO_2). As future research, a heuristic will be developed in order to solve the second sub-problem (allocation of the deliveries to the trucks) where the MIP solver doesn't find solutions and in order to find good feasible solutions within reasonable time when the MIP solver requires to much time in finding a solution. The results given by this heuristic will be compared with those computed by CPLEX. Some of the sub-problems presented in this paper will also be combined together and formulated as a unique MIP. Various alternatives will be compared in order to observe the difference of efficiency of the MIP solver when solving identical instances in terms of quality of the solutions and time needed.

References

[1] C. Archetti, M. W. P. Savelsbergh, and M. G. Speranza. To split or not to split: That is the question. *Transportation Research Part E: Logistics and Transportation Review*, 44(1):114–123, 2008.

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

[3] J. M. Belenguer, M. C. Martinez, and E. Mota. A lower bound for the split delivery vehicle routing problem. *Operations Research*, 48(5):801–810, 2000.

[4] S. Chen, B. Golden, and E. Wasil. The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks*, 49(4):318–329, 2007.

[5] J.-F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J.-S. Sormany. New heuristics for the vehicle routing problem. In *Logistics Systems: Design and Optimization*, pages 279–297. Public Health Service, 2005.

[6] G. Desaulniers. Branch-and-Price-and-Cut for the Split-Delivery vehicle routing problem with time windows. *Operations Research*, 58(1):179–192, 2010.

[7] M. Dror, G. Laporte, and P. Trudeau. Vehicle routing with split deliveries. *Discrete Applied Mathematics*, 50(3):239–254, 1994.

[8] M. Dror and P. Trudeau. Savings by split delivery routing. *Transportation Science*, 23(2):141–145, 1989.

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

[10] P. W. Frizzell and J. W. Giffin. The split delivery vehicle scheduling problem with time windows and grid network distances. *Computers & Operations Research*, 22(6):655–667, 1995.

[11] S. C. Ho and D. Haugland. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*, 31(12):1947–1964, 2004.

[12] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

[13] P. A. Mullaseril, M. Dror, and J. Leung. Split-Delivery routeing heuristics in livestock feed distribution. *The Journal of the Operational Research Society*, 48(2):107–116, 1997.

[14] V. Schmid. *Trucks in Movement: Hybridization of Exact Approaches and Variable Neighborhood Search for the Delivery of Ready-Mixed Concrete*. PhD thesis, Universität Wien, 2007.