Integer linear programming models for a cement delivery problem

Alain Hertz

Département de mathématiques et de génie industriel
École Polytechnique de Montréal
alain.hertz@gerad.ca

Marc Uldry and *Marino Widmer*

Decision Support & Operations Research
Department of Informatics
University of Fribourg
marc.uldry@unifr.ch and marino.widmer@unifr.ch

April 4, 2011

Abstract

We consider a cement delivery problem with an heterogeneous fleet of vehicles and several depots. The demands of the customers are typically larger than the capacity of the vehicles which means that most customers are visited several times. This is a split delivery vehicle routing problem with additional constraints. We first propose a two phase solution method that assigns deliveries to the vehicles, and then builds vehicle routes. Both subproblems are formulated as integer linear programming problems. We then show how to combine the two phases in a single integer linear program. Experiments on real life instances are performed to compare the performance of the two solution methods.

Keywords: vehicle routing, split deliveries, integer linear programs

1 Introduction

In the classical Capacitated Vehicle Routing Problem (CVRP), an homogeneous fleet of vehicles with limited capacity has to serve a set of customers from a single depot with the objective of minimizing the total traveled distance. Each customer has to be visited exactly once and the total demand of the customers visited by a vehicle can not exceed its capacity. An overview of solution methods for the CVRP can be found in [12], [10] and [13].

In the split delivery vehicle routing problem (SDVRP), the restriction that each customer is visited once is removed. The SDVRP has been introduced by Dror and Trudeau (1989) [7] who derived structural properties of optimal SDVRP solutions and empirically showed that allowing split deliveries can lead to substantial cost savings. The same conclusion is obtained in [1]. In [3], a lower bound is proposed for the SDVRP where the demand of each customer is lower than the capacity of the vehicles, and the quantity delivered by the vehicles when visiting a customer is an integer number. Valid inequalities for the SDVRP are described in [6] while branch-and-price and branch-and-price-and-cut algorithms are proposed in [8] and [5]. Tabu search algorithms for the SDVRP are described in [11] and [2]. Frizzell and Giffin (1995) [9] present a mathematical formulation and a heuristic algorithm for the SDVRP with grid network distances and time window constraints. Real applications of the SDVRP are also studied in [14], [15] and [16] and are reviewed in [4].

In this paper, we study a cement delivery problem which can be formulated as a SDVRP with additional constraints. For example, the fleet of vehicles is heterogeneous, there are several depots where cement is available for the customers, and some vehicles may be loaded in advance, before the beginning of their daily activities. A precise description of the problem will be given in the next section. We then describe in Section 3 a two phase method where a set of deliveries is first assigned to each vehicle, and these deliveries are then ordered to build vehicle routes. Each one of the two subproblems is formulated as an integer linear programming problem. In Section 4, we formulate the cement delivery problem as a unique integer linear program that combines the two phases of the first method. Comparisons between the two solution methods are given in Section 5.

2 Problem Definition

A cement supplier company has a heterogeneous fleet V of vehicles that can be used to deliver cement to a set C of customers. Each vehicle $k \in V$ has a capacity Q_k. We consider a set I of orders, each order $i \in I$ being characterized by a non-negative quantity d_i and a customer $c_i \in C$ to which d_i units of cements must be delivered. For a customer $c \in C$, we denote by I_c the subset of orders i with $c_i = c$. The demands of the customers are typically larger than the capacity of the vehicles, which means that most customers must be visited several times to satisfy their demands.

Cement can be supplied either from a central depot of from local depots which typically correspond to railway stations where cement is brought by train. We denote by $D = \{0, \cdots, \mid D \mid -1\}$ the set of depots where 0 denotes the central depot. The company maintains all stock levels at a value that is high enough to ensure that the total demand of the customers can be satisfied. This means that the quantity of cement available at each depot can be considered as unlimited. Some vehicles do not have the required equipment to load cement at the local depots, which means that the cement delivered by these vehicles necessarily comes from the central depot. We denote by V^L the set of vehicles which can load cement from local depots.

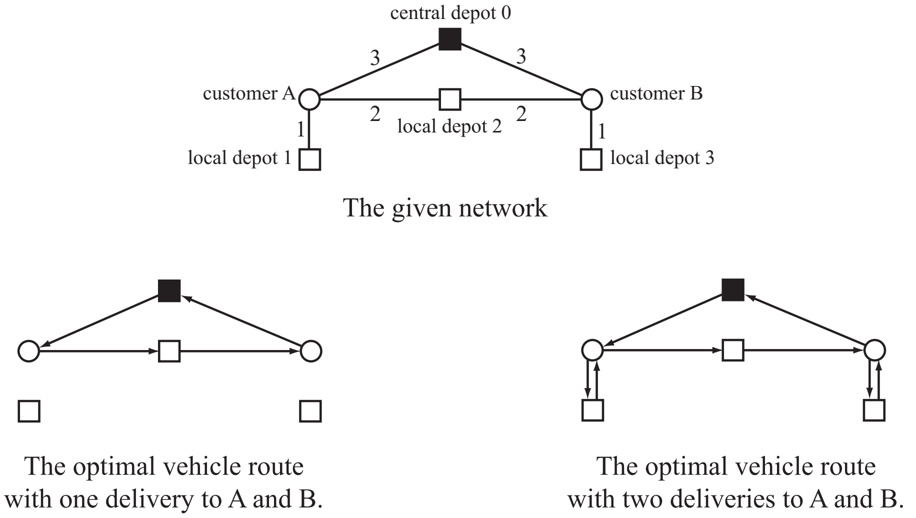
It may occur that the demand d_i associated with some order $i \in I$ is small enough to be supplied with only one delivery. In such a case, a vehicle $k \in V$ with capacity Q_k strictly smaller than d_i should not be used for the delivery. Also, some vehicles cannot transport some types of cement, which explain why they cannot make some deliveries. To take such constraints into account, we denote by V_i the subset of vehicles which can be used to deliver cement for order $i \in I$, and by I_k the subset of orders that vehicle k can handle. Also, some types of cement cannot be loaded at some local depots. We denote by D_i the subset of depots where cement can be loaded for order $i \in I$.

Each vehicle must start and end its daily activity at the central depot. This means that the first delivery of each vehicle is loaded at the central depot, and every vehicle has to drive back to the central depot after its last delivery. No vehicle can transport cement for two different orders at the same time, which means that after a delivery to some customer, the vehicle has to travel to a depot to load cement for the next delivery.

Loading times depend on the depot where cement is loaded, on the type of cement, and on the vehicle used. Unloading times depend on the customer where cement has to be delivered, on the type of cement, and on the vehicle used. We denote by L_{ijk} the time needed by vehicle $k \in V$ to load cement at depot $j \in D$ for order $i \in I$, and by U_{ik} the time it needs to unload that cement at customer c_i. Travel times depend on the vehicle used, on whether or not it is loaded, and are moreover proportional to distances. We denote by TL_k (respectively, TU_k) the time needed by vehicle $k \in V$ to travel one kilometer when it is loaded (respectively, unloaded). Also, we denote by δ_{cj} the distance from customer c to depot j. We suppose that the distances are symmetrical, which means that δ_{cj} is also the distance from depot j to customer c. Each minute spent by a vehicle $k \in V$ for loading, unloading or traveling has a fixed cost F_k. This cost takes into account a tax for carbon dioxide emissions as well as the use of vehicles rented by the company. Some vehicles are possibly not available during portions of the day (for maintenance or other reasons). We denote by A_k the total availability time (in minutes) of vehicle $k \in V$.

For a vehicle $k \in V$ and an order $i \in I_k$, we denote by Δ_{ik} the depot $j \in D_i$ that minimizes $L_{ijk}+\delta_{c_ij}TL_k$. In words, Δ_{ik} is a depot where cement can be loaded by k for order i, and it is chosen in D_i so that it minimizes the sum of the loading time and the travel time to c_i.

When a vehicle $k \in V$ makes multiple deliveries to a customer for a same order $i \in I$, these have to appear consecutively on the vehicle route. It is always optimal, except possibly for the first delivery, to load the required cement at the depot Δ_{ik}. The use of another depot for the first delivery can however shorten the route. This is illustrated with the following example. Assume that the loading times are insignificant and consider a vehicle $k \in V^L$ that has to perform only two deliveries, one to customer A and then one to customer B. Assume that there are three local depots where cement can be loaded for these two deliveries: the local depot 1 is at distance 1 from A and 5 from B, the local depot 2 is at distance 2 from A and B, and the local depot 3 is at distance 5 from A and 1 from B. Also, we assume that the central depot is at distance 3 from A and B. In the shortest vehicle route, customer A receives cement loaded at the central depot while the cement delivered to B is loaded at the local depot 2. If the vehicle has to perform two deliveries to A and two to B, then the second delivery to A is made from the local depot 1 which is the closest to A, while the second delivery to B is made from the local depot 3 which is the closest to B. This is illustrated on Figure 1.



The given network

The optimal vehicle route with one delivery to A and B.

The optimal vehicle route with two deliveries to A and B.

Figure 1.

To save time, some vehicles are 'preloaded' (loaded in advance) at the end of the previous day so that they can start their trip earlier in the next morning, without loosing time with the loading process. Some of these preloads cannot be used for all orders. A possible reason for this can be that a customer has to be delivered very early in the morning and we want to make sure that the preload is used to deliver that customer. Another reason can be that the preload contains a specific type of cement that does not correspond to some orders. We denote by $VP \subset V$ the subset of vehicles that are preloaded and by IP^k ($k \in VP$) the subset of orders $i \in I$ such that vehicle k can start its daily activity with a delivery for i.

Finally, it is required that no more than Nc different vehicles are used to satisfy all orders of a same customer, and that no vehicle loads cement from more than Nd different depots. The cement supplier company can also impose that no more than Nk vehicles are used to perform all deliveries.

A natural objective for this problem is to determine a set of vehicle routes with minimum total cost. The company however possibly has another objective which is, for example, to avoid situations where customers are supplied by more than one vehicle.

3 A Two Phase Method

To solve this cement delivery problem, we first propose a decomposition into two subproblems which are solved sequentially. We first assign deliveries to each vehicle, and then order the deliveries on each vehicle route. Each of these subproblems is formulated as an integer linear programming problem.

3.1 Assignment of deliveries to the vehicles

For every order $i \in I$ and every vehicle $k \in V$, we define the following variables :

$$n_{ik} = \text{the number of deliveries made by vehicle } k \text{ for order } i$$

$$x_{ik} = \begin{cases} 1 & \text{if at least one delivery is made for order } i \text{ by vehicle } k \\ 0 & \text{otherwise.} \end{cases}$$

For every preloaded vehicle $k \in VP$ and every order $i \in IP^k$, we define

$$y_{ik} = \begin{cases} 1 & \text{if preloaded vehicle } k \text{ makes its first delivery for order } i. \\ 0 & \text{otherwise.} \end{cases}$$

For every depot $j \in D$ and every vehicle $k \in V^L$, we define

$$z_{jk} = \begin{cases} 1 & \text{if vehicle } k \text{ makes at least one delivery for an order } i \text{ with } \Delta_{ik} = j \\ 0 & \text{otherwise.} \end{cases}$$

For every customer $c \in C$ and every vehicle $k \in V$, we define

$$v_{ck} = \begin{cases} 1 & \text{if vehicle } k \text{ makes at least one delivery to customer } c \\ 0 & \text{otherwise.} \end{cases}$$

Finally, for every vehicle $k \in V$, we define

$$w_k = \begin{cases} 1 & \text{if vehicle } k \text{ makes at least one delivery} \\ 0 & \text{otherwise.} \end{cases}$$

The objective is to minimize the total time needed to make all deliveries. When a vehicle k delivers cement for some order i, we know that at least $n_{ik}-1$ deliveries are made with cement loaded at depot Δ_{ik}. If $k \in V^L$, the first delivery by k for order i possibly comes from another depot, depending on the location of the previous customer visited by k. But since the sequence of deliveries on each vehicle route is not known yet, we cannot determine at this stage the origin of the first delivery by k for order i. Also, at the end of the n_{ik} deliveries, we don't know yet the next destination of vehicle k. To estimate the total time that vehicle k needs to make all its deliveries, we assume that all deliveries for order i are made from Δ_{ik} and that the vehicle travels back to Δ_{ik} when all its deliveries for order i are accomplished. Hence, for each delivery associated with order i, we sum up the loading time at depot Δ_{ik}, the unloading time at c_i, the loaded travel time from Δ_{ik} to c_i and the unloaded travel time from c_i to Δ_{ik}. All this is in fact only valid for a vehicle $k \in V^L$ that can load cement at local depots. For a vehicle $k \notin V^L$, we use the central depot to compute the exact time (i.e., not an estimation) needed by k for its deliveries. In summary, let

$$R_{ik} = \begin{cases} L_{i\Delta_{ik}k} + U_{ik} + \delta_{c_i\Delta_{ik}}(TL_k + TU_k) & \text{if } k \in V^L \text{ and } \Delta_{ik} \neq 0 \\ L_{i0k} + U_{ik} + \delta_{c_i0}(TL_k + TU_k) & \text{otherwise.} \end{cases}$$

The objective is then to minimize the following function:

$$\sum_{i \in I} \sum_{k \in V_i} F_k R_{ik} n_{ik} \tag{1}$$

We have to consider the following constraints. We first impose that no vehicle $k \in V$ is used for more that A_k minutes. As mentioned above, for a vehicle $k \in V^L$, R_{ik} is only an estimation of the time needed by k to make a delivery for order i. Hence, if the total estimated delivery time of vehicle k is not larger than A_k, it may happen that the real delivery time is strictly larger than A_k. To avoid such situations, we reduce the availability time of all vehicles $k \in V^L$. This is done by bounding the estimation error. More precisely, assume that i is the first order in the route of vehicle k. The distance δ_{c_i0} from the central depot to c_i is possibly strictly larger than the distance $\delta_{c_i\Delta_{ik}}$ used in the estimation. Also, the real loading time at the central depot is L_{i0k} if vehicle k is not loaded in advance (i.e., $k \notin VP$) and 0 otherwise, while the estimated loading time is $L_{i\Delta_{ik}k}$. In summary, assuming that i is the first order on the route of vehicle k, we define the estimation error T^k_{0i} due the first delivery to i as follows:

$$T^k_{0i} = \begin{cases} -L_{i\Delta_{ik}k} + TL_k(\delta_{c_i0} - \delta_{c_i\Delta_{ik}}) & \text{if } k \in VP \\ (L_{i0k} - L_{i\Delta_{ik}k}) + TL_k(\delta_{c_i0} - \delta_{c_i\Delta_{ik}}) & \text{if } k \notin VP. \end{cases}$$

Similarly, if i is the last order on the route of vehicle k, we define the estimation error T^k_{i0} due to the travel back from c_i to the central depot as follows:

$$T^k_{i0} = TU_k(\delta_{c_i0} - \delta_{c_i\Delta_{ik}}).$$

Finally, the third estimation error is on the time separating the last delivery for an order i and the first delivery for the next order i' on the route of vehicle k. Indeed, if i and i' and consecutive orders on the route of vehicle k, and if depot $j \in D_{i'}$ is chosen for the first delivery to $c_{i'}$, we have to add up the unloaded travel time from c_i to j, the loading time at j, and the loaded travel time from j to $c_{i'}$, which gives a total time of $TU_k\,\delta_{c_ij}+L_{i'jk}+TL_k\,\delta_{c_{i'}j}$. This is to be compared with the estimation $TU_k\,\delta_{c_i\Delta_{ik}} + L_{i'\Delta_{i'k}k} + TL_k\,\delta_{c_{i'}\Delta_{i'k}}$ obtained from R_{ik} and $R_{i'k}$. The estimation error due to the choice of j instead of $\Delta_{i'k}$ is then the difference between these two times, and we choose the depot $j \in D_{i'}$ with smallest error. Formally, this third estimation error, denoted $T^k_{ii'}$, is defined as follows:

$$T^k_{ii'} = \min_{j\in D_{i'}}\{TU_k(\delta_{c_ij} - \delta_{c_i\Delta_{ik}}) + (L_{i'jk} - L_{i'\Delta_{i'k}k}) + TL_k(\delta_{c_{i'}j} - \delta_{c_{i'}\Delta_{i'k}})\}.$$

The total estimation error on the time needed by vehicle k to make all its delivery can however not be computed since the sequence of deliveries is not known yet. The following value E^k_1 is an upper bound on the estimation error due to the first delivery performed by vehicle k and to its last travel back to the depot:

$$E^k_1 = w_k(\max_{i\in I_k} T^k_{0i} + \max_{i\in I_k} T^k_{i0}).$$

Also, since there are exactly $\sum_{i\in I_k} x_{ik} - w_k$ pairs or consecutive orders on the route of vehicle k, the following value E^k_2 is an upper bound on the estimation error due to consecutive orders:

$$E^k_2 = \begin{cases} (\sum\limits_{i\in I_k} x_{ik} - w_k) \max\limits_{\substack{i,i'\in I_k\\ i\neq i'}} T^k_{ii'} & \text{if } |I_k| > 1 \\ 0 & \text{otherwise.} \end{cases}$$

In summary, $E^k_1+E^k_2$ is an upper bound on the total estimation error. Hence, by limiting the estimated delivery time of vehicle k to $A_k - E^k_1 - E^k_2$, we know that the real delivery time will not be larger than A_k. Such a reduction of the total availability of vehicle k is however possibly too restricting since the real error can be much smaller that the upper bound. We therefore consider a parameter $\theta \in [0, 1]$ and impose the following constraints: (2) and (3).

$$\sum_{i\in I} R_{ik}n_{ik} \leq A_k - \theta(E^k_1 + E^k_2) \qquad \forall k \in V^L. \tag{2}$$

$$\sum_{i\in I} R_{ik}n_{ik} \leq A_k \qquad \forall k \notin V^L. \tag{3}$$

We then impose constraints (4) to ensure that all demands are satisfied.

$$\sum_{k \in V_i} Q_k n_{ik} \geq d_i \qquad \forall i \in I. \tag{4}$$

Constraints (5) impose that each order $i \in I$ is delivered by vehicles that can perform such deliveries.

$$\sum_{k \notin V_i} n_{ik} = 0 \qquad \forall i \in I. \tag{5}$$

Let $N_{ik} = \lceil d_i / Q_k \rceil$ denote the number of deliveries needed to satisfy order $i \in I$ if all these deliveries are performed by vehicle $k \in V$. Constraints (6) and (7) link variables x_{ik} with variables n_{ik}.

$$n_{ik} \geq x_{ik} \qquad \forall i \in I, \forall k \in V \tag{6}$$

$$n_{ik} \leq N_{ik} x_{ik} \qquad \forall i \in I, \forall k \in V. \tag{7}$$

Constraints (8) impose that each preloaded vehicle $k \in VP$ performs its first delivery for an order $i \in IP^k$ while constraints (9) link variables x_{ik} with variables y_{ik}.

$$\sum_{i \in IP^k} y_{ik} = 1 \qquad \forall k \in VP \tag{8}$$

$$y_{ik} \leq x_{ik} \qquad \forall k \in VP, \forall i \in IP^k. \tag{9}$$

Constraints (10) link variables x_{ik} with variables z_{jk}, while constraints (11) ensure that no vehicle in V^L loads cement from more than Nd different depots. Note that constraints (11) are possibly too restrictive. Indeed, if a vehicle k makes only one delivery for an order i (i.e., $n_{ik} = 1$), then $z_{\Delta_{ik}k} = 1$ while vehicle k possibly does not load any cement from depot Δ_{ik}. Such a situation is however not frequent since the demands of the customers are typically larger than the capacity of the vehicles, which means that most vehicles will perform multiple deliveries for a same order.

$$z_{\Delta_{ik}k} \geq x_{ik} \qquad \forall i \in I, \forall k \in V^L \cap V_i \tag{10}$$

$$\sum_{j \in D} z_{jk} \leq Nd \qquad \forall k \in V^L. \tag{11}$$

Constraints (12) link variables x_{ik} with variables v_{ck} and constraints (13) ensure that no more than Nc vehicles are used to satisfy the demand of a customer.

$$v_{c_i k} \geq x_{ik} \qquad \forall i \in I, \forall k \in V_i \tag{12}$$

$$\sum_{k \in V} v_{ck} \leq Nc \qquad \forall c \in C. \tag{13}$$

Finally, constraints (14) link variables v_{ck} with variables w_k and constraints (15) impose that no more than Nk vehicles are used to make all deliveries.

$$w_k \geq v_{ck} \qquad \forall k \in V, \forall c \in C \tag{14}$$

$$\sum_{k \in V} w_k \leq Nk. \tag{15}$$

To summarize, the problem of assigning deliveries to the vehicles can be solved by minimizing objective (1) under constraints (2)–(15).

The cement supplier company possibly has other objectives such as avoiding situations where more than one vehicle is used to supply the demands of a customer, or minimizing the number of vehicles used to make the deliveries. This can be achieved by minimizing $\sum_{c \in C} \sum_{k \in V} v_{ck} + \lambda \sum_{k \in V} w_k$, where λ is a parameter that gives more or less importance to the minimization of the number of vehicles used. We can then discriminate among solutions with the same objective value by choosing one with minimum toital delivery time. For this purpose, we consider a constant $M = \sum_{k \in V} F_k A_k$ which is obviously larger or equal to the optimal solution of the above integer linear program. We therefore define the following objective function,

$$M(\sum_{c \in C} \sum_{k \in V} v_{ck} + \lambda \sum_{k \in V} w_k) + \sum_{i \in I} \sum_{k \in V_i} F_k R_{ik} n_{ik} \tag{16}$$

and the problem to solve is then to minimize objective (16) under constraints (2)–(15).

3.2 Sequencing the orders on each route

Assuming that a set of deliveries has been assigned to each vehicle, we now show how to sequence the deliveries on each route. This is in fact a traveling salesman problem that has to be solved for each vehicle $k \in V$. Indeed, let S_k denote the set of orders handled by vehicle k (i.e., $n_{ik} > 0$ if and only if $i \in S_k$) and consider $S'_k = S_k \cup \{0\}$, where 0 stands for the central depot. For each ordered pair (a, b) of elements in S'_k, we define

$$s^k_{ab} = \begin{cases} 1 & \text{if } b \text{ is the immediate successor of } a \text{ on the route of vehicle } k \\ 0 & \text{otherwise.} \end{cases}$$

Constraints (17) and (18) impose that every $a \in S'_k$ has an immediate successor and an immediate predecessor on the route of vehicle k.

$$\sum_{\substack{b \in S'_k \\ a \neq b}} s^k_{ab} = 1 \qquad \forall k \in V, \forall a \in S'_k. \tag{17}$$

$$\sum_{\substack{b \in S'_k \\ a \neq b}} s^k_{ba} = 1 \qquad \forall k \in V, \forall a \in S'_k. \tag{18}$$

Every preloaded vehicle $k \in VP$ makes its first delivery for the order $i \in IP^k$ with $y_{ik} = 1$. This is imposed by the following constraints:

$$s_{0i}^k \geq y_{ik} \qquad \forall k \in VP, \forall i \in IP^k. \tag{19}$$

In order to forbid subtours, we now define the following new variables t_a^k for every $k \in V$ and every $a \in S'_k$:

$$t_a^k = \text{position of } a \text{ in the route of vehicle } k$$

with the convention that the central depot is at position 0 while positions $1, \cdots, \mid S_k \mid$ are occupied by the orders in S_k.

Since each route starts at the central depot, we have

$$t_0^k = 0 \qquad \forall k \in V. \tag{20}$$

Constraints (21) are imposed to bound the values of these new variables.

$$0 \leq t_a^k \leq |S_k| \qquad \forall k \in V, \forall a \in S_k. \tag{21}$$

Constraints (22) impose that t_a^k is strictly smaller than t_b^k whenever $s_{ab}^k = 1$. Hence, the orders handled by vehicle k necessarily have consecutive positions on its route.

$$t_a^k - t_b^k + |S_k|\, s_{ab}^k \leq |S_k| - 1 \qquad \forall k \in V, \forall a \in S'_k, \forall b \in S_k, a \neq b. \tag{22}$$

As mentioned in Section 3.1, function (1) only estimates the total delivery time of the vehicles. To minimize the real delivery time, we have to minimize the estimation error which is defined with the following function (see Section 3.1 for the definition of T_{ab}^k):

$$\sum_{k \in V} \sum_{a \in S'_k} \sum_{\substack{b \in S'_k \\ a \neq b}} T_{ab}^k s_{ab}^k. \tag{23}$$

In summary, the problem of finding the best sequence of orders on each route can be solved by minimizing objective (23) under constraints (17)–(22). We can of course solve $|V|$ independent problems, one for each k.

4 Solving the problem in one phase

We now show how the two subproblems defined in Sections 3.1 and 3.2 can be combined into one integer linear programming problem. For this purpose, we use exactly the same variables as those defined in the previous sections. In order to link the two suproblems we have to impose that an order i has a successor and a predecessor on the route of vehicle k if and only if $i \in S_k$. This is easily imposed by linking the s^k_{ab} variables with the x_{ik} and w_k variables. For this purpose we replace constraints (17) and (18) by the following constraints:

$$\sum_{\substack{b \in I \cup \{0\} \\ a \neq b}} s^k_{ab} = x_{ak} \qquad \forall k \in V, \forall a \in I \tag{24}$$

$$\sum_{\substack{b \in I \cup \{0\} \\ a \neq b}} s^k_{ba} = x_{ak} \qquad \forall k \in V, \forall a \in I \tag{25}$$

$$\sum_{i \in I} s^k_{0i} = w_k \qquad \forall k \in V \tag{26}$$

$$\sum_{i \in I} s^k_{i0} = w_k \qquad \forall k \in V. \tag{27}$$

Since we don't know in advance the set S_k of orders handled by vehicle k, we replace constraints (21) and (22) by constraints (28) and (29) , where $n = |I|$:

$$0 \leq t^k_a \leq n \qquad \forall k \in V, \forall a \in I \tag{28}$$

$$t^k_a - t^k_b + n\, s^k_{ab} \leq n - 1 \qquad \forall k \in V, \forall a \in I \cup \{0\}, \forall b \in I, a \neq b. \tag{29}$$

Note that constraints (29) impose that the orders on the route of vehicle k have different but not necessarily consecutive positions. For example, if a problem has 10 orders and vehicle k handles only 3 of them, they may have positions 2, 7 and 9. The fact that these positions are different is however sufficient to forbid subtours.

As already mentioned, by adding the total estimated delivery time (1) and the total estimation error (23), one gets the real total delivery time. For vehicle k, this real delivery time, denoted α_k, is defined as follows:

$$\alpha_k = \sum_{i \in I} R_{ik} n_{ik} + \sum_{a \in I \cup \{0\}} \sum_{\substack{b \in I \cup \{0\} \\ a \neq b}} T^k_{ab} s^k_{ab}.$$

We can now replace constraints (2) and (3) by the following constraints :

$$\alpha_k \leq A_k \qquad \forall k \in V \tag{30}$$

and the new objective function that combines objectives (1) and (23) is simply

$$\sum_{k \in V} F_k \alpha_k. \tag{31}$$

In summary, the cement delivery problem can be solved by minimizing objective (31) under constraints (4)–(15), (19), (20), and (24)–(30).

Again, if the company wants to avoid situations where more than one vehicle is used to supply the demands of a customer, or if it is interested in minimizing the total number of vehicles used for the deliveries, we can use the following objective function which is similar to function (16):

$$M(\sum_{c \in C} \sum_{k \in V} v_{ck} + \lambda \sum_{k \in V} w_k) + \sum_{i \in I} \sum_{k \in V_i} F_k \alpha_k. \tag{32}$$

5 Computational experiments

In order to evaluate the performance of the solution methods described in Sections 3 and 4, we have run experiments using the CPLEX interactive optimizer 12.2.0.0 on a PC (Core 2 Duo 3.00 GHz, 4 GB RAM).

The cement supplier company has provided us with the data of ten instances which correspond to real life orders spread over two workweeks. For each instance, we indicate in Table 1 the number $|I|$ of orders, the number $|C|$ of customers, and the number $|VP|$ of preloaded vehicles. There are 14 local depots (railway stations) in addition to the central depot.

Table 1: characteristics of the test set.

Instance	$\|I\|$	$\|C\|$	$\|VP\|$
1	24	21	0
2	29	21	1
3	31	25	0
4	25	22	0
5	29	27	3
6	33	27	2
7	28	25	1
8	27	24	5
9	29	25	3
10	20	18	5

We have set $Nd = 2$ in constraints (11) and $Nc = 10$ in constraints (13). We have chosen the objective (16) for the first subproblem of the two phase method and the similar objective (32) for the one phase method. In both cases, we have fixed $\lambda = 0.1$. We have removed constraint (15) (which is equivalent to set $Nk = \infty$) since the total number of vehicles used to make the deliveries is minimized in the considered objective functions.

As explained in Section 3, parameter θ in the two phase method helps to avoid situations where a vehicle $k \in V^L$ is used for more than A_k minutes. While the use of $\theta = 1$ in constraints (2) ensures that there will be no overtime, such a setting can be too restrictive. We have run the two phase method with θ varying between 0 and 1, with step 0.1. The total overtime (i.e, $\sum_{k \in V} max\{0, \alpha_k - A_k\}$ is represented in Figure 2, using box-and-whisker plots. More precisely, for each value of θ, we consider the 10 total overtimes (one per instance) produced by the two phase method and we construct a box between the lower quartile Q_1 and the upper quartile Q_2, with a solid line drawn across the box to locate the median. A value smaller than $(Q_1 - 1.5(Q_2 - Q_1))$ or larger than $(Q_2 + 1.5(Q_2 - Q_1))$ is defined as an outlier and is plotted using an empty circle. Two 'whiskers' are attached at the top and at the bottom of the box: the lower (upper) whisker ends at the minimal (maximal) value that is not an outlier.

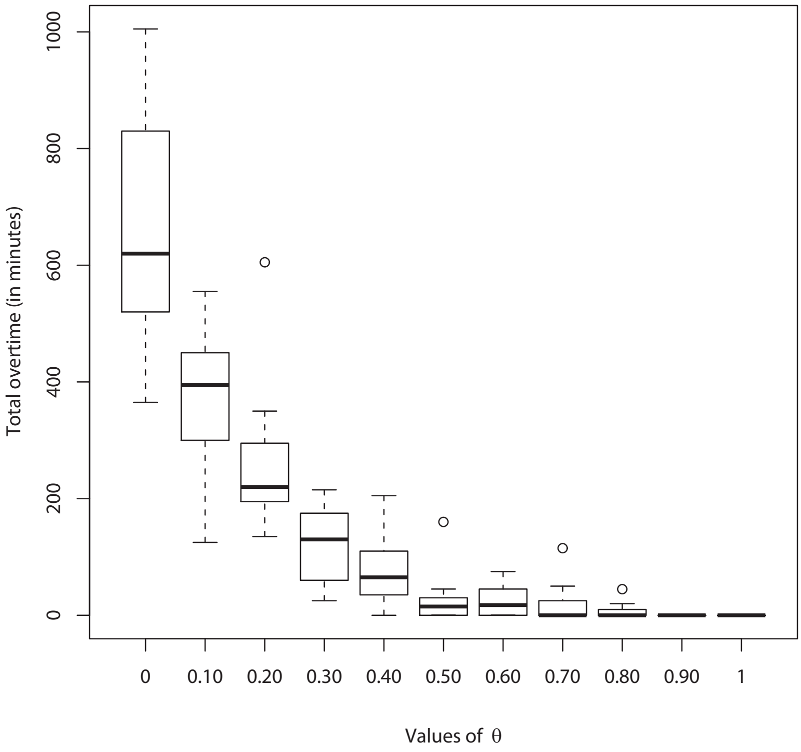


Figure 2: relation between θ and the total overtime.

We observe that with $\theta \geq 0.9$ we get no violation, while the total overtime reaches 1'000 minutes when $\theta = 0$. The company has informed us that trade union rules allow an overtime of 120 minutes per week per driver. It turns out that if one driver is assigned to each vehicle, then the solutions obtained with $\theta \geq 0.5$ satisfy this condition. Table 2 provides more details on the overtime obtained when using the two phase method with $\theta = 0.5$. For each instance, we indicate the total (i.e. $\sum_{k \in V} max\{0, \alpha_k - A_k\}$) and the maximum overtime. The total overtime of 160 minutes for instance 3 corresponds to the outlier in Figure 2.

From now on, we denote $2P_{1.0}$ and $2P_{0.5}$ the two phase method used with $\theta = 1$ and $\theta = 0.5$, and we denote 1P the one phase method. Comput-

Table 2: overtime with $\theta = 0.5$.

Instance	Total	Maximum
1	0	0
2	0	0
3	160	70
4	5	5
5	45	45
6	25	25
7	0	0
8	30	20
9	0	0
10	25	25

ing times are compared in Table 3. For $2P_{1.0}$ and $2P_{0.5}$, we only consider the first subproblem since the optimal solutions of the second subproblem (see Section 3.2) were all found in less than one second. For each instance, we indicate the time (in seconds) needed by CPLEX for finding an optimal solution and proving its optimality. If no proof of optimality could be obtained after 6 hours of computation, we indicate the gap between the value of the best feasible solution and the best lower bound. In four cases (namely, instances 3, 5, 6 and 8), 1P was stopped before the time limit of 6 hours because the computer ran out of memory, and the corresponding computing times are shown in italic. In one case (instance 6), no feasible solution was found before 1P had to be stopped.

Table 3: Computing times and gaps to optimality

	$2P_{1.0}$		$2P_{0.5}$		1P	
Instance	CPU	Gap	CPU	Gap	CPU	Gap
1	22	0.00%	511	0.00%	>6 hours	0.63%
2	9'957	0.00%	1'390	0.00%	>6 hours	0.93%
3	6	0.00%	1'899	0.00%	*12'412*	8.95%
4	16	0.00%	>6 hours	0.12%	>6 hours	0.95%
5	>6 hours	0.03%	588	0.00%	*8'997*	7.59%
6	19	0.00%	8'112	0.00%	*5'332*	-
7	7	0.00%	8	0.00%	>6 hours	1.09%
8	855	0.00%	455	0.00%	*12'705*	0.88%
9	>6 hours	0.13%	1'117	0.00%	>6 hours	0.87%
10	756	0.00%	448	0.00%	>6 hours	5.40%

We observe that the two phase method has produced feasible solutions for the ten instances and most of these solutions (8 for $2\mathrm{P}_{1.0}$ and 9 for $2\mathrm{P}_{0.5}$) are proven optimal. By analysing the output file generated by CPLEX we have observed that the 8 optimal solutions found by $2\mathrm{P}_{1.0}$ and 8 of the 9 optimal solutions found by $2\mathrm{P}_{0.5}$ have been produced in less than 15 minutes, the hardest and longest task being to prove the optimality. For comparison, the one phase method has generated 9 feasible solutions, and no proof of optimality could be obtained in 6 hours of computation. This however does not means that $2\mathrm{P}_{1.0}$ or $2\mathrm{P}_{0.5}$ should be preferred to 1P. Indeed, the value of the optimal solution found by $2\mathrm{P}_{1.0}$ is an upper bound on the optimal value that 1P tries to determine. This is also the case for $2\mathrm{P}_{0.5}$ if it delivers a solution satisfying constraints (30). However, if $2\mathrm{P}_{0.5}$ generates a solution with overtime (and this is the case for instances 3, 5, 6, 8 and 10), then the optimal value that 1P is looking for is possibly strictly larger than the value of the solution generated by $2\mathrm{P}_{0.5}$.

In Table 4, we compare the quality of the solutions produced by $2\mathrm{P}_{1.0}$, $2\mathrm{P}_{0.5}$ and 1P. We also report the value of the solutions implemented by the cement supplier company (columns CSC). For each instance, we indicate the total value of the variables v_{ck} (which is the main objective to minimize), the total number $\sum_{k\in V} w_k$ of vehicles used (which is the second most important objective), and the total delivery time (column $\sum F_k\alpha_k$).

We first analyse the main objective which is to minimize $\sum_{c\in C}\sum_{k\in V} v_{ck}$. In other words, the cement supplier company would ideally like to deliver all orders of a customer with a unique vehicle. We observe that 1P has found 5 such ideal solutions (instances 1, 2, 7, 8, 9), while the two phase method has found 2 (with $\theta = 1$) and 3 (with $\theta = 0.5$) such solutions. For comparison, the solutions implemented by the cement supplier company are very far from such an ideal situation. Indeed, if we consider for example instance 6, we can observe that $\sum_{c\in C}\sum_{k\in V} v_{ck} = 38$ while the number $|C|$ of customers is equal to 27. We also observe that 1P typically produces lower values than $2\mathrm{P}_{0.5}$, which is always better than $2\mathrm{P}_{1.0}$ for this objective. There are however 3 exceptions where $2\mathrm{P}_{0.5}$ has found better values than 1P: one is for instance 6 where 1P has not found any feasible solution, and the two other exceptions are for instances 3 and 5 where 1P ran out of memory while the gap to optimality was larger than 7%.

We next analyse the second objective which is to use as few vehicles as possible. In 3 cases (instances 2, 5, 6), the solution implemented by the company uses less vehicles than the one produced by $2\mathrm{P}_{1.0}$. However, for this second objective, $2\mathrm{P}_{0.5}$ is always strictly better than $2\mathrm{P}_{1.0}$ and CSC, and 1P is always better than $2\mathrm{P}_{0.5}$, with the exception of instance 6 where 1P has not found any feasible solution, and instance 3 where the solution produced by $2\mathrm{P}_{0.5}$ has a total overtime of 160 minutes (see Table 2), which makes it possible to use one less vehicle.

Table 4: solution values.

			$\sum v_{ck}$				$\sum w_k$				$\sum F_k \alpha_k$			
Instance	$\|I\|$	$\|C\|$	CSC	$2P_{1.0}$	$2P_{0.5}$	1P	CSC	$2P_{1.0}$	$2P_{0.5}$	1P	CSC	$2P_{1.0}$	$2P_{0.5}$	1P
1	24	21	25	23	23	21	13	13	11	10	5’940	5’195	4’860	4’550
2	29	21	32	24	23	21	16	17	14	13	7’430	6’585	6’300	5’710
3	31	25	33	25	25	27	16	16	13	14	7’445	7’075	6’525	6’765
4	25	22	27	25	24	24	15	15	13	12	7’260	6’685	6’145	5’645
5	29	27	32	30	28	29	17	19	16	15	8’180	8’270	7’660	7’175
6	33	27	38	27	27	-	17	19	15	-	7’485	7’320	6’630	-
7	28	25	31	26	25	25	17	17	15	14	7’195	6’580	6’160	5’735
8	27	24	33	27	25	24	17	15	12	12	6’920	6’035	5’720	5’435
9	29	25	35	29	26	25	17	17	14	13	7’860	7’365	6’510	6’025
10	20	18	26	20	20	20	17	16	13	13	7’440	6’375	5’870	5’690

The comparison of the total delivery times goes in the same direction as the second objective. Indeed, the solutions implemented by the company have a strictly larger total delivery time than those produced by $2P_{1.0}$, with the exception of instance 5, where this time is larger by 90 minutes (1.1%). Notice that the gain for instance 10 is 1’065 minutes, which corresponds to an improvement of 14.3%. The total delivery time of the solutions found with $2P_{0.5}$ are always strictly better than those obtained with $2P_{1.0}$ or implemented by the company. The gain even reaches 1’570 minutes for instance 10, which corresponds to an improvement of 21.1%. Here again, 1P always produces the best values, except for instance 6 where no feasible solution was found, and for instance 3 where the violation by $2P_{0.5}$ of constraint (30) made it possible to use 13 instead of 14 vehicles and to reduce the total delivery time by 240 minutes.

The following comments summarize the above analysis and point out the advantages and drawbacks of each method.

- $2P_{1.0}$ always produces feasible solutions which are typically better than those implemented by the company. Optimal solutions are generally obtained within a few minutes. There are however some exceptions. For example, the solution found by $2P_{1.0}$ for instance 5 requires strictly more vehicles than the solution implemented by the company (19 instead of 17) and the total delivery time is also strictly larger (by 90 minutes). We can however observe a decrease of 2 units (from 32 to 30) in the main objective $\sum_{c \in C} \sum_{k \in V} v_{ck}$. Such a decrease in the main objective even reaches 11 units (34.4%) for instance 6, and the reduction of the total delivery time (third objective) reaches 14.3% for instance 10.

- $2P_{0.5}$ is strictly better than $2P_{1.0}$. The total delivery time of the solutions generated by $2P_{0.5}$ is in average 14.8% smaller than that implemented by the company and the number of required vehicles is reduced by up to 5 units (instance 8). These good solutions are all obtained within a few seconds or minutes. The only drawback of $2P_{0.5}$ is that it produces solutions with a possible overtime, which means that constraints (30) are not necessarily satisfied. We have however observed that the total violation typically does not exceed the limit imposed by trade union rules, and such a violation can induce a significant improvement in some components of the objective function. A typical example is instance 3 where $2P_{0.5}$ has produced a solution that requires strictly less vehicles than the other methods (second objective) and with the smallest total delivery time (third objective).

- 1P ran out of memory in 4 out 10 cases, has not obtained any proof of optimality, but has produced 9 feasible solutions of very high quality. When comparing the solutions of the company with those produced by 1P, we observe a decrease of the total delivery time by more than 23% for instances 1, 2, 9 and 10 (the average decrease is 19.8%), while the gain in the main objective $\sum_{c \in C} \sum_{k \in V} v_{ck}$ reaches 11 units (34.4%) for instance 2, and the reduction in the number of required vehicles reaches 5 units (29.4%) for instance 8. Instance 3 is the unique exception where 1P is slightly worse than $2P_{0.5}$, but this is because $2P_{0.5}$ was allowed to generate solutions with overtime. The unique drawback of 1P is that the integer linear program is much harder to solve than that of the two phase method, and it may happen that no feasible solution is produced within a few hours (which is exactly what happened with instance 6).

The ten instances used in our experiments are representative of the daily workload of the company and the above results clearly show that both the one phase and the two phase methods provide better solutions than those generated by the company.

6 Conclusion

We have proposed integer linear programming models for a cement delivery problem. We have first proposed a two phase method that decomposes the problem into two subproblems (assignment of the deliveries to the vehicles and sequencing of the orders on each route) and solves them sequentially. We have then shown how the two subproblems can be combined into one integer linear programming problem.

Computational experiments performed using CPLEX have shown that the feasible solutions produced by the one phase and the two phase method with $\theta = 0.5$ are always better than the solutions implemented by the cement supplier company. The two phase method with $\theta = 0.5$ was able to prove the optimality of 9 of the 10 feasible solutions it has produced, and 8 of them were obtained in less than 15 minutes. The one phase method ran out of memory without producing any feasible solution in one case, and has generated the best feasible solutions in all other cases.

Finally, we want to mention that while the proposed mathematical models have been designed for the solution of a cement delivery problem, they can also be used in other similar contexts such as the transportation of logs from forest sites to customers.

References

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

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

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

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

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

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

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

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

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

[10] Gendreau, M., Laporte, G. and Potvin, J.-Y. (2002). Metaheuristics for the Capacitated VRP. In: Toth, P., Vigo, D. (eds.) The vehicle routing problem, SIAM Monographs on Discrete Mathematics and Applications, pp. 129-154. SIAM, Philadelphia.

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

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

[13] Laporte, G. and Semet, F. (2002). Classical Heuristics for the Capacitated VRP. In: Toth, P., Vigo, D. (eds.) The vehicle routing problem, SIAM Monographs on Discrete Mathematics and Applications, pp. 109-128. SIAM, Philadelphia.

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

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

[16] Siermsma, G. and Tijssen, G.A. (1998). Routing helicopters for crew exchanges on off-shore locations. *Annals of Operations Research* 76: 261–286.