# Column Generation based Solution for a Tactical Inventory Routing Problem

Sophie Michel, François Vanderbeck

# Column Generation based Solution for a Tactical Inventory Routing Problem *

S. Michel and F. Vanderbeck

University Bordeaux 1, Mathematic Institute of Bordeaux, 351 Cours de la Libération, F-33405 Talence

and INRIA Futurs (Bordeaux), research team RealOpt

Emails: sophie.michel/francois.vanderbeck@inria.fr

September 3, 2007

The inventory routing problem combines the issue of organizing product delivery or pick-up, and managing the inventory at each customer site. Our application deals with planning product pick-ups, customer inventory being emptied on each visit. We assume a deterministic production rate. The inventory costs are limited to those resulting from the transportation model. In this tactical model, the objective is to minimize the fleet size and to ensure that routes are geographically clustered. The planning must be repeated over the time horizon with constrained periodicity. We develop a truncated branch-and-price algorithm combined with rounding and local search heuristics that yield both primal solutions and dual bounds. On a large scale test problem coming from industry, we obtain a solution within 6.5% deviation of the optimal. A rough comparison with the industrial practice shows a 10% decrease in number of vehicles as well as in travel distance. The key to the success of the approach is the use of a state-space relaxation technique in formulating the master program to avoid the symmetry in time. The paper also includes a short review of column generation based heuristics.

*Keywords:* Inventory Routing, Branch-and-Price-and-Cut, Primal Heuristic, Symmetry.

*Research Report INRIA-00169311

# Introduction

In the mid-1980s, research work began on integrating the inventory control to vehicle routing in an effort to better manage an important segment of the supply chain. The Inventory Routing Problem (IRP) consists in designing routes for deliveries or pick-ups that integrate issues of inventory management at customer sites. Three decisions have to be made: (i) when to serve a customer; (ii) how much to deliver (resp. pick-up) to (resp. from) a customer when it is served; and (iii) which delivery (resp. pick-up) routes to use. Many variants are discussed in the literature (see the classification in Table 1 of Section 2 for a summary). Federgruen and Simchi-Levi [31] discuss the first studies on IRP, while a recent survey is provided by Cordeau et al. [25].

The application that motivates our study concern the design of routes for collecting a single product from customers who accumulate it in their stock. Our aim is to provide a decision aid algorithm for tactical planning. One needs first to dimension the size of the vehicle fleet needed to collect the product. Then, among equal fleet size solutions, one can consider several side objectives. Our industrial partner raised issues that are more important to him that the traditional minimization of travel distances (the latter is left for the operational planning model). One issue is to arrive at a solution where cluster of visited points that are assigned to a vehicle are gathered in a compact geographical area. Hence, we attempt to minimize dispersion from a cluster center. Another issue concerns the ease of the implementation of the planning. Hence, we impose a cyclic planning of constrained periodicity over the time horizon. The tactical solution should serve as a target in operational planning so as to make the latter less myopic. Therefore, it is desirable to build robust solutions that resist to stochastic variation of the accumulation rate at the operational level. A standard practice is to account for such variations by reserving buffer space in the vehicles and/or the customer stocks, and/or by overestimating the filling rates.

The model considered here was, to the best of our knowledge, not explicitly considered in the literature, but many variants were. Most of the existing approaches tend to make restrictive assumptions (such as assuming a fixed partition policy as explained below) or to adopt a hierarchical optimization scheme where planning is decided before routing. Most approaches are heuristics with no warranty on the deviation to optimality and are specific to the problem variant. We develop a truncated branch-and-price-and-cut algorithm combined with rounding and local search heuristics that yields both primal so-

lutions and dual bounds and hence allows to bound the deviation to optimality. Periodic plans are generated for vehicles by solving a multiple choice knapsack subproblem. The issues related to the construction of the customer planning are dealt with in a master program. We were confronted with the issue of symmetry in time that naturally arises in building a cyclic schedule (cyclic permutations along the time axis defines alternative solutions). Central to our approach is a state-space relaxation idea that allows to avoid this symmetry. Our algorithm provides solutions with reasonable deviation to optimality for large scale problems (260 customers, 60 time periods, 10 vehicles) coming from industry.

The paper is organized as follows. Section 1 describes our problem and specifies our assumptions. We then make a review of the existing literature (Section 2). In Section 3, we outline our decomposition approach. The Dantzig-Wolfe reformulation eliminates the symmetry in vehicle indexing (vehicles are identical) but not the symmetry in time. Hence, we model an average behavior by considering a single aggregate variable measuring how many times a specific route and associated pick-up quantities is used over all possible starting dates. In Section 4 we compare this aggregate formulation to the discrete time formulation. Section 5 presents the specific features of our column generation approach to solve the master LP. Cutting planes and partial branching are used to improve the dual bounds as presented in Section 6. Heuristics based on column generation give primal bounds. A short review of such methods is given in Section 7. Our implementation is presented in Section 8. Finally, Section 9 presents our results on an industrial test problem and compares them with the current industrial practice before concluding the paper.

# 1 The Tactical Planning Problem

A fleet of vehicles is devoted to collecting a single product from geographically dispersed sites. Each site has its own accumulation rate and stock capacity. At the tactical planning level, filling rates at collection points are seen as deterministic. Because of technical constraints, the stock must be fully emptied on each pick-up, so the inventory management rule is what is known as an "order-up-to-level" policy. Thus, the collected quantities can be normalized in number of periods that have passed since the last visit. The customer stock capacity implies a maximum interval between two visits. The stock management costs reduce to the transportation costs.

The problem is to design a planning and associated routes for collecting the product from customers over a given time horizon. In fact, we consider an infinite time horizon

but we search for a periodic solution. We restrict the solution space by imposing that route periodicities are selected from a restricted set $P$. This implies a bound, $T$, on the time horizon beyond which the solution is repeated. $T$ is the least common multiple (LCM) of the periodicities in $P$. For our study, we take $P = \{1, 2, 3, 4, 5, 6\}$[1] and, hence, $T = 60$. For each route, we must select its periodicity $p \in P$ and its first occurrence, i.e. its starting date, $s \leq p$. Then, the solution is $H$ periodic where $H$ is the LCM of the periodicities used in the solution. $T$ acts as the maximal length of the regeneration cycle, i.e. $H \leq T$. Hence, $T$ can be seen as the finite time horizon that results from the restriction on the periodicities. The planning requirements boil down to ensuring that the stocks produced on each period, $t = 1 \ldots, T$, are picked-up in some route.

The exact routing of vehicles is considered as an operational issue. In the tactical planning model, we minimize fleet size and attempts "to regionalize" vehicle routes. In the sequel, we abusively use of the term "route" to denote the planning restricted to a specific vehicle and a specific period: a route is defined by the cluster of visited customer sites and the specific quantities that are collected on each visited point (their sum cannot exceed the vehicle capacity); a vehicle can only cover one route per period. The objectives are to minimize the maximum number of routes used per period (i.e. the size of the vehicle fleet), and to achieve some form of regional clusterization. We define below a cluster cost that estimates the regional compacity of a cluster. The objective of our model is a weighted sum of the number of used vehicles per period and the average cluster cost per period.

The input to our model are the periodicity set, $P$, the resulting time horizon bound, $T$ (periods are indexed by $t = 1, \ldots, T$), and a set of customers, $N$, where 0 stands for the garage where routes start from and 1 is the depot where vehicles are unloaded (let $N' = N \backslash \{0, 1\}$ be its restriction to true customer sites). For each customer, $i \in N'$, we know the maximum time lag between two visits, $t_i^{\max}$, the normalized inventory filling rate per period, $r_i$, and the collect time, $f_i$. Moreover, we know the distance matrix between sites: $\{d_{ij}\}_{(ij)\in N\times N}$ denotes the shortest distances between sites (in time unit). Finally, we have $V$ identical vehicles of capacity $W$ (we assume $V \geq \lceil \frac{\sum_i t_i^{\max} r_i}{W} \rceil$).

A cluster of visited customer sites, $S \subset N$, has a cost defined as the sum of the distances of the visited points to the cluster center. The latter is selected as one of the

[1] In our real-life data, the frequency of visits that is induced by the maximum interval between two visits falls naturally into set $\{1, 2, 3, 4, 5, 6\}$ for 60% of the customers.

visited points (it is the seed of the route). Thus, each route is associated to a star in the graph of customer points[2]. To be more accurate, the cost of a cluster, $S$, that is built around a seed, $k \in S$, includes a setup cost that is defined as length of a shortest path from the garage to the depot passing by this seed, plus the seed collect time, i.e., the fixed cluster cost is $c_k = d_{0k} + d_{k1} + f_k$, where $d_{0k}$, $d_{k1}$ are the travel times between the garage 0 or the depot 1 and the site $k$. Then, the connection cost for customer $i$ to the seed $k$ is the cost of the best insertion of site $i$ in the path garage-seed-depot plus its collect time, i.e., $c_{ik} = d_{ik} + min\{d_{0i} - d_{0k}, d_{1i} - d_{1k}\} + f_i$ [3]. Minimizing this measure favor the grouping of customers who are geographically close to each other or who are on the path garage-seed-depot. The planning constraints will induce the formation of clusters that group customers sharing the same frequency of collect.

A compact formulation of the problem can be derived in terms of the following variables: $x_{i\ell vps} = 1$ *iff* customer $i \in N'$ is collected a quantity equal to the accumulation of its stock over $\ell \leq t_i^{\max}$ periods by vehicle $v$ that performs a route of periodicity $p \in P$ and starts in $s \leq p$; $y_{vps}$ indicates the use of vehicle $v$ for a route of periodicity $p \in P$ that starts in $s \leq p$; $z_{ikvps} = 1$ *iff* customer $i \in N'$ is visited by a route of seed $k \in N'$ performed by vehicle $v$ that has periodicity $p \in P$ and starts at date $s \leq p$ (note that $z_{kkvps} = 1$ *iff* the customer $k$ is a seed); and $Vmax$ is the maximum number of vehicles that are used in a period. We make use of an indicator vector, $\delta^{ps}$, that tells us whether a vehicle is used by a plan indexed by $(p, s)$ in a given period $t$ (if $y_{vps} = 1$, vehicle $v$ is used in periods $t \in \{s, s+p, s+2\,p, \ldots\}$):

$$\delta_t^{ps} = \begin{cases} 1 & \text{if } \exists m \in I\!N \text{ such that } s + m * p = t, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Similarly, we define an indicator matrix, $\delta^{\ell ps}$, saying whether the stock produced by a customer $i \in N'$ in a given period $t \in \{1, \ldots, T\}$ is collected by a plan indexed by $(l, p, s)$ (if $x_{i\ell vps} = 1$, the production of customer $i$ is collected for periods $t \in \{s - \ell + 1, \ldots, s - 1, s, s - \ell + 1 + p, \ldots, s - 1 + p, s + p, \ldots\}$):

$$\delta_{it}^{\ell lps} = \begin{cases} 1 & \text{if } \exists m \in I\!N \text{ and } \tau \in \{0, \ldots, \ell - 1\} \text{ such that } s + m * p - \tau = t, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

[2] Christofides and Eilon [23] have shown that the expected route length $D_o$ is monotonously linked to the sum of radial distances $D_r$ between the customers and a given center $r$ : $D_o \approx B\sqrt{a}\sqrt{D_r}$, where $B$ is a constant and $a$ is the side of the square containing the customers. Thus, our radial cost can also be understood as an approximation of travel times.

[3] This cost structure was used by Fisher and Jaikumar [32] for their heuristic to solve the VRP.

Thus, our compact formulation is:

$$\min \quad Vmax + \alpha \sum_{v,p,s} \frac{1}{p} \sum_{i,k} (c_k z_{kkvps} + c_{ki} z_{ikvps}) \tag{3}$$

$$\sum_{\ell,v,p,s} \delta_{it}^{\ell ps} \, x_{i\ell vps} = 1 \qquad \forall i \in N', t = 1, \ldots, T \tag{4}$$

$$\sum_{k \in N'} z_{ikvps} = \sum_{\ell} x_{i\ell vps} \qquad \forall i \in N', v, p, s \tag{5}$$

$$z_{ikvps} \leq z_{kkvps} \qquad \forall i \in N', k \in N', v, p, s \tag{6}$$

$$\sum_{i \in N', \ell} \ell \, r_i \, x_{i\ell vps} \leq W y_{vps} \qquad \forall v, p, s \tag{7}$$

$$\sum_{v,p,s} \delta_t^{ps} \, y_{vps} \leq Vmax \qquad \forall t = 1, \ldots, T \tag{8}$$

$$x_{i\ell vps} \in \{0, 1\} \qquad \forall i, \ell, v, p, s \tag{9}$$

$$y_{vps} \in \{0, 1\} \qquad \forall v, p, s \tag{10}$$

$$z_{ikvps} \in \{0, 1\} \qquad \forall k, i, v, p, s \tag{11}$$

$$Vmax \in I\!N \tag{12}$$

where constraints (4) ensure that, for each customer, the stock produced in each period is collected, constraints (5) enforce the relation between variables $x$ and $z$, constraints (6) define cluster centers, constraints (7) enforce the bound on the vehicle capacity, and constraints (8) determine the fleet size. The objective (3) minimizes the number of vehicles used and includes the average cluster cost per period weighted by a coefficient $0 \leq \alpha < 1$ which balances both terms.

# 2 Literature Review

To clarify the position of our application in the diversity of variants of the Inventory Routing Problem (IRP), we present a classification in Table 1. This review is focused on deterministic approaches and deliberately omit other papers (as [40], f.i., where the stochastic IRP is formulated as a Markov decision process). We then provide a brief summary of the solution approaches that have been used. In the classification of Table 1, we take 3 basic criteria to characterize IRPs: the number of products, the length of the time horizon (we specify $T$ when the horizon involves multiple base periods), and the type of demand. A more comprehensive classification could be designed that combines all the variants of the VRP with the variants of the inventory management problem for customers. Moreover, some studies consider a production management problem for suppliers in addition to the customer inventory control and routing model.

Regarding, the number of products, we note that in most of the problems encountered in the literature, one transports a single product type. When several product types are considered, one has two situations: either the products can be loaded in the same truck, or a truck has different tanks and each tank received a single product type (f.i., in the problem of delivering fuel to gas stations [6, 41, 48]). In the multiple product case, one can often return to the the single product situation: either one can aggregate products in a single type with an average consumption rate [34], or decouple the problem into independent product problems [14, 53] (if products are loaded in different trucks and customers are duplicated for each product), or consider indistinctive products [8] (if products are loaded in the same trucks and customers are duplicated for each product).

In the classification according to the time horizon, we distinguish finite versus infinite horizon. The horizon, $T$, is divided into base periods which often stand for a day, but it could be a hour [1, 34] or a week as in our application. In some studies, one optimizes over a single period ($T = 1$) at the time and reiterates the process on each period. This hierarchical optimization procedure, although sub-optimal, allows to adjust to operational data (receiving real data every morning as in [20] or from the latter routes as in [35]). When dealing with multiple periods ($T > 1$), a classical approach is to use a rolling horizon: the planning is optimized on $T$ periods, but the operational routes are implemented only for the first $\tau < T$ periods and the process reiterates from $\tau + 1$ using updated data. (This is applied, f.i., in special cases as in [53], where each customer needs a single visit during $T$).

Global optimization approaches over the $T$ periods can be distinguished into operational models that assume an initial situation [2, 4, 10, 22, 30] or tactical planning models that aim at providing a periodic solution [34]. An infinite horizon is only used at the tactical level where one minimizes the average costs on the long term. In this case, searching for an optimal solution can be very difficult and solution can become quickly impracticable where the horizon of the regeneration cycle is huge [4]. To overcome this drawback, one often restricts the solution space using replenishment strategies as for example:
($i$) direct shipping: a tour delivers a single customer; or
($ii$) zero inventory policy: a customer is replenished if and only if its inventory is down to zero (with this policy, however, the structure of the solution space remains too complex

[4] Consider a vehicle with unlimited capacity and 12 customers $i = 1, \ldots, 12$, where customer $i$ must be visited every $i$ periods. If a route visits all 12 customers in period 1, it will be re-used only in period $27720 = \text{LCM}(\{1, \ldots, 12\})$.

- **1 product** :
  - **horizon with a single period**:
    - **deterministic demands**: [20];
    - **stochastic demands**: [35] (liquid propane);
  - **rolling horizon**:
    - **on-line demands**: [8] ($T$= 2 to 5 days while $\tau$ = 1 day, gases: oxygen, nitrogen...);
    - **deterministic demands**: [26] ($T$=2 periods while $\tau$ =1 period), [53] ($T$= 5 days while $\tau$ = 1 day, gas), [14, 15] ($T$= 30 days while $\tau$ = 2 days, gas);
    - **stochastic demands**: [38, 7] ($T$=10 days while $\tau$ = 5 days, liquid propane);
  - **horizon with multiple periods**:
    - **deterministic demands**: [34] ($T$= 6 days, supermarket chain, periodic solution), [4, 10, 21, 22] ($T$= 30 days, ship routing of ammonia);
    - **stochastic demands**: [30] ($T$= 5 days);
  - **infinite horizon**:
    - **deterministic demands**: [1, 3, 12, 19, 52];
- **multiple products,** loaded in **same tank**:
  - **infinite horizon**:
    - **deterministic demands**: [27] (frozen products);
    - **stochastic demands**: [45];
- **multiple products,** loaded in **different tanks**:
  - **horizon with a single period**:
    - **deterministic demands**: [6, 41, 48] (gas stations);
  - **horizon with multiple periods**:
    - **deterministic demands**: [2] (liquid bulk product by ship);

Table 1: Classification of previous studies on the Inventory Routing Problem

to permit a reasonable search, hence the following restriction is more common);
(*iii*) the fixed partition policy [12]: the set of customers is partitioned into disjoint sets and each set is served separately, i.e., whenever a customer of the set is served, all the

customers in that set must be served. Some extensions of the fixed partition policy are proposed in the literature. In [3], the demand of a customer can be split in several clusters. In [1], the route of a cluster is split into several sub-tours between which the vehicle returns to the depot for re-loading;
($iv$) power of two policy [27]: replenishment times are multiples of a power of two; and
($v$) periodic review policy [45]: a periodicity (restricted to be a integer multiple of a base period) and a replenishment level is set for each customer.

When dealing with a short time horizon at the operational level as in [8], the data are regularly updated and the route construction takes into account an heterogeneous fleet, time windows, ... Whereas, at the tactical level, the studies often simplify the problem by considering an unlimited number of vehicles [3, 52], making restrictive assumptions such that the fixed partition policy [1, 3, 12, 34], or ignoring the storage capacity of the customers [1, 12].

Our third criteria for classification is the consumption rate (or the filling rate). It can be deterministic or stochastic, stationary or time dependent (f.i., [34] assumes a deterministic time-varying demand). The stochastic case is the closest to reality: the risk that a customer falls in stock-out exists. At the operational level, when a customer is in a stock-out situation, he must be replenished by an urgent delivery at a high cost. To take such risk into account, a probability of stock-out is defined. This probability permits to add an error term to average rate [35], or to compute the mean amount of demand in a stock-out and the associated penalties [45] (at the tactical level, one could compute these quantities for all time intervals).

Using a deterministic model is either motivated by the desire to simplify the problem, or by the assumption that variation in consumption rate are small, or it results from having taken into account the stochastic character by way of safety threshold: for each customer, $i$, a maximum interval between two visits is defined such that the probability to be in stock-out does not exceed a given probability $p_i$ [52]; a safety stock is considered [27], or a buffer space can be reserved in trucks [34]. Finally, another way to restrict the problem to a deterministic model is to consider online demand. For instance, [8] defines lower and upper limits on the amount of product that must be delivered to each customer in each time period. Then, customers with a high level of variability, give a telephone call to establish their exact inventory level. Stocks can also be monitored in real-time using sensors. In other applications, users can manage risks with an interactive interface

to change schedules. For these online problems, [8, 53] develop heuristics to re-optimize the planning daily in response to contingencies such as extra demand.

Our classification hides some elements characterizing the variants of the IRPs. Let us mention just a few that are related to our application. [10, 30, 52] assume that, on each delivery, the quantity delivered is such that the maximum level of inventory is reached (this is called the order-up-to-level policy). Then, the quantity delivered is defined by the schedule and is not a decision variable, as in our model. Another consideration is whether a vehicle is limited to execute a single route per period, or whether it can cover a sequence of several consecutive routes with a visit to the depot in between each sub-tour before going back to the garage (this is called a multi-tour or a rotation) as in [1, 35]. The objective function varies also depending on the stand point. When the decision-maker is the transporter, he minimizes transportation cost. If customers are in the same company as the transporter, inventory cost are minimized too. If the revenue depends on the quantity delivered, then the objective is to maximize total profit [8, 20]. When the number of vehicles is unlimited, the average vehicle requirement can be minimized [52].

In view of the problem variants reported above, we can summarize the literature by saying that none of the previous study deals with the same model as ours under the same assumptions. The closest study is probably the paper of Webb and Larson [52]. They deal with a tactical planning problem where the stock management policy is an order-up-to-level policy assuming deterministic consumption rate. They show that the fixed partition policy combined with order-up-to-level policy is very restrictive. They partition their customers in subsets and define for each partition a set of routes that are repeated periodically. Their heuristic uses the concept of Clarke Wright savings. In our case, the customers are not partitioned a priori and our approach is based on an exact method.

It is also interesting to consider the size of the data for problems that are dealt with in the literature. For example, the number of customers is limited to 15 in the ammonia distribution by ship in [22] and it goes up to 3000 in liquid propane distribution in [35]. Let us also look at the number of customers that are visited by a single vehicle (a measure of the difficulty of the vehicle route generation subproblem): in [34], 2 supermarkets are replenished by a vehicle; in [6, 8], 4 customers or gas stations are served; while in our application, around 10 points are collected. Note that a route can take a few hours [53] or several days as in ship routing [22].

Let us conclude this literature review with an overview of the solution approaches that have been used to tackle these complex problems. Most are heuristics with no warranty on the deviation to optimality and are specific to the problem variant. However some asymptotic analysis of delivery policies are provided in [3, 12, 19]. Exact approaches have been used to solve very small problems: [21, 22] use a Branch-and-Price algorithm, [4] a Branch-and-Cut algorithm. The existing heuristic approaches tend to adopt a hierarchical optimization scheme where planning is decided before routing (f.i., see [15]). In a first step, customers to be visited and/or replenishment volumes are defined for each period in trying to take into account the impact on transportation costs (by generating a set of possible vehicle routes a priori [8, 15], or by solving a TSP as in [35], or estimating a fixed cost linked to routes as in [30]). In the second step, routes are developed for each period (typically by an insertion heuristic followed by an exchange heuristic). These two steps can be re-optimized iteratively by passing feedback information as done in [45]. With the fixed partition policy, these two steps are fitted into each other because the choice of clusters also defines the routes (one needs to solve a TSP for each cluster) and the planning (one can solve a shortest path problem for each cluster as in [34], or solve a EOQ model as in [3, 12]).

Most methods somehow make use of a concept of decomposition with classical subproblems such as a TSP, a VRP, a bin packing, a knapsack problem, or an EOQ model. The above two-step method amounts to decomposing the problem with the inventory management on the one hand and the routing problem on the other hand. An explicit Dantzig-Wolfe decomposition is used by [53] (the master problem manages inventory and subproblems manage the routes), and by [21, 22] (the problem is decomposed into a ship route subproblem for each ship and a harbor inventory subproblem for each harbor). [8, 20] base their heuristics on a Lagrangian relaxation which decomposes the problem into knapsack subproblems.

# 3 A Dantzig-Wolfe Decomposition approach

In the line of the previous literature, a decomposition approach seems natural for the tactical inventory routing planning problem. We apply the Dantzig-Wolfe reformulation principle [51] to compact formulation (3-8), dualizing the planning constraints (4) and the fleet size constraints (8). The problem decomposes into a master program taking care of the inventory planning issues on one hand and subproblems defining routing clusters on the other hand (a subproblem solution defines a route specifying the visited customers

with the quantities picked-up at each site expressed in number of periods worth, its periodicity and its starting date).

Once the periodicity, $p$, of a route is fixed, as well as its starting date, $s$, and its seed, $k$, the problem of selecting the members of the cluster and associated picked-up quantities reduces to a variant of the multiple choice knapsack problem (noted MCKP). Let $x_{i\ell} = 1$ *iff* customer $i$ is in the cluster and the quantity that is collected is its production of $\ell$ periods. The associated profit is $p_{i\ell}$. The MCKP takes the form:

$$\max \sum_{i,\ell} p_{i\ell} x_{i\ell} \tag{13}$$

$$\sum_{\ell} x_{k\ell} = 1 \tag{14}$$

$$\sum_{\ell} x_{i\ell} \leq 1 \qquad \forall\, i \neq k \tag{15}$$

$$\sum_{i,\ell} \ell\, r_i\, x_{i\ell} \leq W \tag{16}$$

$$x_{i\ell} \in \{0,1\} \qquad \forall\, i, \ell. \tag{17}$$

Constraint (14) determines the collected quantity for seed $k$, constraints (15) choose at most one collected quantity for the other customers, and constraint (16) is the knapsack constraint enforcing the bound on vehicle load.

Let $\{(c^q, x^q, p^q, s^q)\}_{q \in Q}$ be the enumerated set of periodic routes, $q$, that are defined as a solution, $x^q$, to the above knapsack subproblem for a fixed triplet $(k, p, s)$ along with its cost, $c^q$, its periodicity, $p^q$, and its starting date, $s^q$. From the information given by $(x^q, p^q, s^q)$, one can generate the indicator $\delta^q$ similarly to (1-2). Thus, the inventory routing problem can be reformulated as the master program:

$$Z^d_{IP} = \min\ Vmax + \alpha \sum_{q \in Q} \frac{c^q}{p^q}\, \lambda_q \tag{18}$$

$$\sum_{q} \delta^q_{it} \lambda_q \geq 1 \qquad \forall i \in N',\ t = 1, \ldots, T \tag{19}$$

$$\sum_{q} \delta^q_{t} \lambda_q \leq Vmax \qquad \forall t = 1, \ldots, T \tag{20}$$

$$\lambda_q \in \{0,1\} \qquad \forall\, q \in Q \tag{21}$$

$$Vmax \in I\!N \tag{22}$$

where $\lambda_q = 1$ *iff* periodic route $q$ is used, while $Vmax$ is the maximum number of vehicles used in a period. $Z^d_{LP}$ denotes the associated LP relaxation value. The variables, $\lambda_q$,

and associated columns are generated dynamically in the course of the LP optimization procedure using a column generation approach.

The above formulation (18-22) avoids the symmetry in the vehicle indexing $v$ that was present in compact formulation (3-8), but it still suffers from a symmetry in $t$: equivalent solutions can be defined that differ only by a permutation in the choice of starting dates. In search for integer solutions, one might enumerate these equivalent solutions. Moreover, the instability in dual variables $\pi$ and $\sigma$ associated respectively to constraints (19) and (20) for every period $t$ is harmful for the convergence of the column generation procedure used to compute dual bounds. To avoid these drawbacks, we aggregate periods and model an average behavior. Technically speaking, we implement a state-space relaxation in the space of the columns: aggregating all columns that differ only by their starting dates, $s^q$, we project our column space as follows:

$$\{(c^q, x^q, p^q, s^q)\}_{q\in Q} \quad \rightarrow \quad \{(c^r, x^r, p^r)\}_{r\in R}.$$

To each column, $r \in R$, is associated a set, $Q(r)$, of columns, $q \in Q$, such that $r$ is the projection of $q$:

$$Q(r) = \{q \in Q : c^q = c^r, x^q = x^r, p^q = p^r, s^q \in \{1, \ldots, p^r\}\} .$$

While the former formulation is referred to as the *discrete time master* problem, the reformulation obtained after performing this mapping is called the *aggregate master*. It takes the form:

$$Z^a_{IP} = \min \quad Vaver \quad + \quad \alpha \sum_{r\in R} \frac{c^r}{p^r}\,\lambda_r \tag{23}$$

$$\sum_{r\in R,l} \frac{\ell}{p^r}\, x^r_{i\ell}\,\lambda_r \quad \geq \quad 1 \qquad\qquad \forall\, i \in N' \tag{24}$$

$$\sum_{r\in R} \frac{1}{p^r}\lambda_r \quad \leq \quad Vaver \tag{25}$$

$$\lambda_r \quad \in \quad I\!N \qquad\qquad \forall\, r \in R \tag{26}$$

$$Vaver \quad \in \quad I\!N, \tag{27}$$

where $\lambda_r$ is the total number of times that a vehicle uses periodic route $r$ ($\lambda_r = \sum_{q\in Q(r)} \lambda_q$), and $Vaver$ is the average number of vehicles used per period ($Vaver \leq Vmax$). Constraints (19) are replaced by (24): each route covers a fraction of the aggregate demand.

Constraints (20) are replaced by (25): each route uses a fraction of a vehicle (the same fraction in each period, in this average model). The dual values, $\pi_i$, associated to constraints (24) represent now the average collect cost for customer $i$. $Z^a_{LP}$ denotes the associated LP relaxation value. In a column generation solution approach of the master LP, the pricing core subproblem takes the form (13-17) as for the discrete master formulation, but one does not have to enumerate on each possible starting date anymore, as dual reward are time independent.

# 4 Comparing discrete and aggregate master program

We show that discrete and aggregate master program have the same optimal LP solution, but the solution of the aggregate master by column generation is much faster (see Table 2). Hence, we use the aggregate master to compute dual bounds. However, from an integer solution point of view, both formulations are not equivalent: the aggregate formulation is a relaxation of the problem. Hence, the discrete time formulation remains useful for computing primal bounds through heuristics.

Let us first illustrate the mapping between discrete and aggregate master solutions on a simple example. Consider a problem with two customers and a solution involving two aggregate routes. Route A collects the production of 2 periods from customer 1 every 2 periods. Route B collects the production of 3 periods from customer 2 every 3 periods. In short notation, we have

- route A: $0 - 1(2) - 0$ with $p_A = 2$,
- route B: $0 - 2(3) - 0$ with $p_B = 3$,

where the digit in the parentheses is the collected quantity. The $LP$ solutions of both formulations can be represented in the form of the following table:

| $\lambda_r$ in the aggregate form. | $\lambda_q$ in the discrete form. | | t1 | t2 | t3 | t4 | t5 | t6 |
|---|---|---|---|---|---|---|---|---|
| $\lambda_A = 1$ | $\lambda_{A_1} = \frac{1}{2}$ | $s_{A_1} = 1$ | ■ | ✓ | ■ | ✓ | ■ | ✓ |
| | $\lambda_{A_2} = \frac{1}{2}$ | $s_{A_2} = 2$ | ✓ | ■ | ✓ | ■ | ✓ | ■ |
| $\lambda_B = 1$ | $\lambda_{B_1} = \frac{1}{3}$ | $s_{B_1} = 1$ | ■ | ✓ | ✓ | ■ | ✓ | ✓ |
| | $\lambda_{B_2} = \frac{1}{3}$ | $s_{B_2} = 2$ | ✓ | ■ | ✓ | ✓ | ■ | ✓ |
| | $\lambda_{B_2} = \frac{1}{3}$ | $s_{B_3} = 3$ | ✓ | ✓ | ■ | ✓ | ✓ | ■ |

For the aggregate formulation, each route is taken once. In the discrete formulation, $p$ discrete time columns are associated to each aggregate route, one for each starting date.

The value $\lambda_q$ of each column is $\frac{1}{p^q}$. In the right part of the table, we indicate, along with the column value, the periods where the vehicle is used by a ■ sign and we mark by a ✓sign the period for which the associated customer demand is collected by the route.

With the intuition of this example, it is clear that every aggregate LP solution, $\{\hat{\lambda}_r\}_r$, of (23-27) can be translated into a discrete time solution, $\{\hat{\lambda}_q\}_q$, to (18-22): the mapping,

$$\hat{\lambda}_q = \frac{1}{p^r}\hat{\lambda}_r \quad \forall r, q \in Q(r) \ , \tag{28}$$

defines a solution such that $\hat{V}max = \hat{V}aver$ and $\hat{Z}^d_{LP} = \hat{Z}^a_{LP}$. Note that an alternative discrete solution ($\overline{\lambda}$, $\overline{V}max$) other that the above symmetric mapping (where $\hat{V}max = \hat{V}aver$) could yield $\overline{V}max > \hat{V}aver$, and therefore an increase cost value. Hence, restricting the discrete LP-solution space to time symmetric solution is not suboptimal. The reverse mapping is trivial. Simply set

$$\hat{\lambda}_r = \sum_{q \in Q(r)} \hat{\lambda}_q \quad \forall r \ . \tag{29}$$

Then, $\hat{V}aver \leq \hat{V}max$ and therefore $\hat{Z}^a_{LP} \leq \hat{Z}^d_{LP}$. From the above mappings, we can conclude that if the discrete master solution is LP optimal for (18-22), then we must have $\hat{V}aver = \hat{V}max$ (otherwise, use transformation (29) followed by (28) to reduce $\hat{V}max$). Thus, we have shown that

**Proposition 4.1** *The LP optimal solution of the discrete master (18-22) can be transformed in an LP optimal solution of the aggregate master (23-27) with the same cost* $Z^{d*}_{LP} = Z^{a*}_{LP}$ *and vice versa.*

To evaluate the comparative advantage of the aggregate master over solving the discrete master LP by column generation, we have carried-on comparative tests on real and randomly generated instances. In Table 2, instances named "IND8" and "IND27" are extraction from real-life industrial data with respectively 8 and 27 customers (results are averages on multiple such extractions). Instances named "RAND100" are random instances with 100 customers imitating the real problem for the order of magnitude of filling rates $r_i$'s and maximal time interval between visits $t_i^{\max}$'s. The customer coordinates are generated according to 3 schemes: urban, rural or mixed. According to a scheme, a region is cut up into squares with specified probability that a customer falls in these squares. Once the square area to which a customer belong has been randomly generated, its coordinates within the square are generated according to an uniform distribution. Travel

times are then assumed to be proportional to Euclidean distances. (We make these random instances available on our web site [37].) Table 2 reports the number of generated columns, "Col", and the overall time spent in computing the LP optimal solution, "Time". All computational times are on a PC bi-pro. Xeon 3GHz, 2Go. A limit of 1 hour is set. Missing inputs correspond to the case where this time limit is reached. On the RAND100, using the discrete formulation increase the computing time by a factor of at least 15.

| | discrete form. | | aggregate form. | |
|---|---|---|---|---|
| instances | Col | Time | Col | Time |
| IND8 (average over 5 inst.) | 71 | 1s75t | 20 | 25t |
| IND27 (average over 10 inst.) | - | >1h | 123 | 5s |
| RAND100 (average over 6 inst.) | - | >1h | 701 | 3m52s |

Table 2: Numerical comparison of LP solution computing effort

Let us now show that both formulations are not equivalent from an integer solution point of view. Some aggregate integer solutions do not have their feasible counterpart in the discrete formulation as illustrated by the following examples. Let us first return on the previous example where one had an integer solution to the aggregate master using $Vaver = 1 \geq \frac{1}{2} + \frac{1}{3}$ vehicles. One can observe that no matter how one chooses the starting dates, there is always one period where both routes require a vehicle; thus, there is no feasible integer solution to the discrete problem using only 1 vehicle. Consider for another illustration a problem where customer 1 that has $t_1^{\max} = 3$ is visited by 2 routes:

- route A: $0 - \cdots - 1(3) - \cdots - 0$ with $p_A = 6$, and
- route B: $0 - \cdots - 1(2) - \cdots - 0$ with $p_B = 4$.

The routes are presented in the following table from the point of view of customer 1 for fixed starting dates ($s_A = 5$ and $s_B = 2$ respectively) – we make use of the same notation as above:

| | | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 | t11 | t12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| route A | $\delta_{1t}^A$ | | | ✓ | ✓ | ■ | | | | ✓ | ✓ | ■ | |
| route B | $\delta_{1t}^B$ | ✓ | ■ | | | ✓ | ■ | | | ✓ | ■ | | |

Assume the aggregate solution has $\lambda_A = \lambda_B = 1$. Although these starting dates avoid conflicts for vehicle use, the production of customer 1 is not properly collected. In fact, there is no integer discrete planning using these 2 routes as any selection of starting dates would leave some demands unsatisfied while over-covering others. In conclusion, even

though the aggregate integer solution can sometimes be translated into a feasible solution for the discrete formulation, this cannot be guaranteed as the aggregate formulation defines a relaxation of the problem. Hence, we work with the discrete time formulation when it comes to constructing primal integer solutions.

# 5 Solving the Aggregate Master LP

We briefly present the column generation procedure that we use to solve the aggregate master LP detailing only the application specific features. We initialize the master formulation with artificial columns associated to constraints (24): artificial column $i$ is defined by the unit vector $e_i$ (the $i$th column of identity matrix) for $i = 1, \ldots, |N'|$. Its cost defines an initial upper bound on the dual value $\pi_i$ (see [50]). We run a dual heuristic to obtain meaningful upper bounds on dual variable values, $\pi_i$. These bounds are then use as initial cost for the artificial columns. ([13] shows the impact on the convergence of the column generation procedure of an intelligent initialization that reduces the heading-in effect as it called in [50].) In solving the master LP, we perform a combined phase 1 and phase 2: if artificial variables remain in the $LP$ optimal solution, their cost is multiplied by 1.8 and the column generation starts again. However, if we fail to eliminate artificial columns from the master solution after several cost increases, we perform a pure phase 1, keeping only the artificial columns in the objective function.

The dual heuristic goes as follows. In an "ideal" solution, each customer $i$ is collected at its optimal frequency $per_i = max\{p \in P, p \leq t_i^{\max}\}$; a vehicle uses its full capacity and collects on average $n = \lceil \frac{W}{(\sum_i r_i \, t_i^{\max})/|N'|} \rceil$ customers that are close neighbors. More realistically, we assume for our dual heuristic that a cluster contains customers who are more spaced out geographically: we assume that a customer $i$ can be connected to one of the $2\,n$ nearest neighbors. Thus, we estimate the contribution of customer $i$ to the inter-customer travel cost to be $\frac{\sum_{j \in N_i} d_{ij}}{2\,n}$ where $N_i$ is the set of the $2\,n$ closest neighbors to customer $i$. To this cost, we must add the fixed collect cost, $f_i$, and an estimate of the garage-depot return trip: customer $i$ contribution to the latter is estimated by $\frac{d_{0i}+d_{1i}}{n}$. Considering that, at $LP$ optimality, $\sum_i \pi_i^* = Vaver^* + \alpha \sum_r \frac{c^r}{p^r} \lambda_r^*$, the average collect cost per period, $\pi_i$, should also include a fixed cost for vehicle use. The latter is derived by estimating $Vaver^*$: if we assume that clusters are ideally made of customers sharing the same frequency, $p \in P$, and that vehicles are devoted to one frequency group, we can derive that they would be $\frac{m(p)}{n}$ clusters of periodicity $p$, where $m(p)$ is the number of customers $i$ with $per_i = p$. Then, $\frac{m(p)}{p\,n}$ vehicles would be required to cover customers

of ideal frequency equal to $p$. Thus, on average, the fixed vehicle cost attributed to a customer can be estimated by $\frac{\sum_p(\frac{m(p)}{p\,n})}{|N'|}$. In summary, our estimator of the contribution of customer $i$ to the total cost is

$$\frac{\sum_p(\frac{m(p)}{p\,n})}{|N'|} + \alpha\,\frac{f_i + \frac{d_{0i}+d_{i1}}{n} + \frac{\sum_{j\in N_i} d_{ij}}{2\,n}}{per_i}\,. \tag{30}$$

As these $\pi_i$ estimates are optimistic, we use as initial values for the $\pi_i$ upper bounds, the value given by (30) multiplied by a factor 1.2.

Once the master is initialized with the above artificial columns, a standard column generation procedure follows. In search for the smallest reduced cost columns, we feed the dual information $\pi$ and $\sigma$ to the pricing problem solver. The latter iterates on each periodicity $p$ and possible seed $k$, solving the associated core multiple choice knapsack problem (13-17) having set $p_{i\ell} = \ell\pi_i - \alpha c_{ki}$ for each $\ell \leq \min\{t_i^{\max}, p\}$. The oracle for the multiple choice knapsack is the dynamic program of Pisinger [44]. The associated solution value is denoted $\zeta_{pk}(\pi)$. The reduced cost of the optimal column $r^*$ is $\overline{c}(\pi,\sigma) = \min_{p\in P, k\in N'} \frac{1}{p}(\sigma + \alpha c_k - \zeta_{pk}(\pi))$.

Some preprocessing is performed to speed up the process. If $\pi_k \leq 0$, then site $k$ cannot be a seed in an optimal solution. For a fixed $i$ and $k$, $x_{i\ell}$ can be set to zero if $\ell\pi_i \leq \alpha c_{ki}$. For a fixed $k$ and $p$, a cut off value on $\zeta_{pk}(\pi)$ is defined from the best reduced cost value encountered on previous pairs $(k,p)$, denoted $\overline{c}_{\text{best}}$: we want $\frac{1}{p}(\sigma+\alpha c_k - \zeta_{pk}(\pi)) \leq \overline{c}_{\text{best}}$, thus $\sigma + \alpha c_k - p\,\overline{c}_{\text{best}}$ defines an lower bound on $\zeta_{pk}(\pi)$ (in particular, if the knapsack problem LP relaxation does not satisfy this bound the problem can be cutoff). In practice, the enumeration of the multiple choice knapsack subproblems stops as soon as a column with negative reduced cost is found (if none such column is found, the optimality is proved). In our numerical tests, we noted that stopping the enumeration of the pricing problem as soon as a column with negative reduced cost is found (instead of searching the best column at each iteration) divides the time by a factor 3.4. Hence, we always use this strategy. To increase the chances to quickly generate good columns, we enumerate periodicities $p$ from the largest down to the smallest (in the $LP$ solution, columns with larger periodicities are more likely to be used). The seeds, $k$, are sorted by decreasing ratio of estimated profit ($= \pi_k * \min\{p, t_k^{\max}\}$) over estimated cluster cost ($= \frac{d_{0k}+d_{1k}+\sum_{i\in N_k} c_{ik}}{2\,n+2}$). A post-optimization improves the returned column: the seed selection is re-optimized and the smallest periodicity $p = \max_i\{\ell : x_{i\ell} = 1\}$ is computed (it can be shown that the master LP solution can be restricted to columns having minimal periodicity).

# 6 Adding Cuts and Performing Partial Branching

Aggregate master LP dual bounds can be slightly improved using a cutting plane procedure. We derive cuts from constraints (24) using a rounding procedure. To illustrate the sort of fractional solution that we aim to cut off, consider the following example. Assume a customer $i$, with $t_i^{\max} = 5$, is visited by a single route of periodicity 6 which covers $\frac{5}{6}$ of its demand. To cover the average demand, this route is taken 1.2 times in the $LP$ solution. An integer solution should use such route at least twice or it should cover the residual demand with another route. A valid inequality to cut this solution is:

$$\sum_{r,\ell:\ \ell=p^r} x_{i\ell}^r \, \lambda_r + \frac{1}{2} \sum_{r,\ell:\ \ell \neq p^r} x_{i\ell}^r \, \lambda_r \geq 1,$$

saying that, to cover any two consecutive periods, one needs either one route which covers all demand or two routes. Such cuts can be generalized into

$$\sum_{r,\ell:\ h\ell \bmod p = 0} \frac{\ell}{p} \, x_{i\ell}^r \, \lambda_r + \sum_{r,\ell:\ h\ell \bmod p \neq 0} (\lceil \frac{h\ell}{p} \rceil - \frac{h\ell}{p}) \, x_{i\ell}^r \, \lambda_r \geq 1 \quad \forall i \in N',\ h \in I\!N \ . \qquad (31)$$

These inequalities can be proved valid: they are obtained from (24) using a super-additive function (following Proposition 4.1, page 229, in [42]) and the equality $\sum_r h\frac{\ell}{p} x_{i\ell}^r \lambda_r = h$ that derives from (24) in its equality form. The super-additive function takes the form:

$$F_\gamma; I\!R \to I\!R; F_\gamma(d) = \lfloor d \rfloor + \frac{(d - \lfloor -d \rfloor - \gamma)^+}{1 - \gamma}$$

with parameter $\gamma$ chosen such that $0 \leq \gamma = 1 - \epsilon < 1$ and $(.)^+ := \max\{0, .\}$ (the super-additivity is proved in Proposition 4.7, page 233, in [42]). Note that when $t_i^{\max} = 1$, (31) are dominated by (24) since, when $h \bmod p \neq 0$, $\lceil \frac{h}{p} \rceil - \frac{h}{p} \geq \frac{1}{p}$. Similarly, when $h = T$, (24) dominates (31), and when $h = T-1$, (24) is equivalent to (31). Moreover, constraints (31) are $T-$periodic in $h$. Thus, we consider cuts (31) for integer $h$ ranging from 1 to $T-2$. As their number is polynomial, separation can be completed by enumeration. One can stop as soon as a violated cut is found. To have a better chance to find violated cut early in the process, the enumeration is made according to some observations. In practice, the inequalities with small values of $h$ are more likely to be violated, just as the ones concerning customers $i$ with a large $t_i^{\max}$ but that is not equal to $Pmax$. After adding a cut, we return to the column generation procedure. The structure of the pricing problem does not change, as only the profits $p_{i\ell}$ are affected. In order to keep the feasibility of the master after adding cuts, a global artificial variable is used. Its cost is an estimation of solution cost (we set it equal to the sum over customer $i$ of estimators (30)). The contribution of

these cuts to dual bound improvement is limited as shown in our numerical experiments below.

To further improve dual bounds, we perform a partial enumeration scheme: we branch only on variable $Vaver$. Given the structure of our objective that focuses on vehicles use, this branching has an important impact on the bound. Assume $Vaver = \beta \notin I\!N$ in the root node aggregate master LP solution, we define two branches

$$\text{(Node 1)} \quad Vaver \leq \lfloor \beta \rfloor \qquad \text{or} \qquad Vaver \geq \lceil \beta \rceil \quad \text{(Node 2)} . \tag{32}$$

In node 1, the branching constraint is very restricting: this branch can often be proved infeasible. In node 2, $Vaver$ is typically integer as a result of the branching constraint and the cost can sometime increase significantly.

To evaluate the impact of cuts and partial branching on the dual bound, we have made comparative tests on the real and randomly generated instances of similar to those of Section 4. We have 5 instances extracted from real-life data with 60 customers on average, this group is named "IND60", and 2 bigger instances with 172 and 157 customers, named "IND172" and "IND157". Moreover, we use the 10 random instances with 100 customer named "RAND100" that are the same instances as those generated for Section 4.

On this test bed, the dual bound improvements observed by adding cut are small (less than 2%). However, cuts change the structure of the $LP$ solution: in the $LP$ solution before adding cuts, the clusters are all 6-periodic, whereas clusters have periodicity of 1 up to 6 in the $LP$ solution after adding cuts. In the dynamic cut generation procedure, on average only 7.63% of the valid inequalities (31) are added to the formulation. On the other hand, the improvements obtained through partial branching can get bigger (depending on the instance and, more specifically, on the fractional part of $Vaver$), it ranges from less than 1% up to more than 15%. To evaluate these bound improvements, we compute the gap to a primal solution (obtained as explained in Section 8). In Table 3, we present the gap obtained at the root of the branch-and-price tree, "gap-root", the gap to the bound obtained after adding cut, "gap-cut", and the gap to the bound obtained after partial branching, "gap-br", as well as combining the latter two, " gap-br-cut" (we call on the cutting plane procedure only at node 2 after rounding up $Vaver$). The root node dual bound is improved by 12.42% when using both cuts and partial branching (which amounts to a 58.5 % improvement in the gap on average).

| Name | gap-root | gap-cut | gap-br | gap-br-cut |
|---|---|---|---|---|
| av RAND100 | 22.81 | 21.31 | 10.37 | 9.24 |
| av IND60 | 26.38 | 25.51 | 12.51 | 11.83 |
| IND172 | 16.46 | 15.59 | 8.65 | 7.89 |
| IND157 | 15.40 | 14.39 | 5.83 | 5.11 |
| **av 17 inst** | **23.05** | **21.80** | **10.63** | **9.67** |

Table 3: Dual bound improvements obtained from cutting planes and partial branching

We also attempt to evaluate the computational burden of adding cuts by comparing computational times. The bulk of the time (around three quarter of the time) is spent in the re-optimization of the master by column generation after each round of additional cuts. However, we observed on our test bed that the average time spent in the cutting plane procedure (including re-optimization of the master) is divided by 3 when we return the first violated cut found instead of the more violated cut (keep in mind our specific order in which we test cut violation). Furthermore, when we add 10 cuts during the same iteration (instead of one at the time), we divide again the time spent in the cutting plane procedure by 1.35. This time is further divided by 1.18 if we solve the pricing problem exactly only every 20 iterations.

When branching on $Vaver$, proving the infeasibility of Node 1 can be very time consuming too. To control this, we limit the number of artificial variable cost increases to 3, after which we perform a pure phase 1. In the Table 4, we show how the computing time is shared between the different component of our dual bound computation. The overall time spent in computing our best dual bound (using cuts and partial branching) varies from 10 minutes for the smallest instances up to more than 2 hours for the largest. The columns of Table 4 indicate the average percentage of that time spent in the pricing problem, "PP", in solving the restricted master, "RM", in the cutting plane generation procedure (which includes separation and re-optimization), "CP" , in the separation procedure alone, “Sep”, and in the different nodes: the root of the branch-and-price tree, "N0", the branch where $Vaver$ is rounded down, "N1", and the branch where $Vaver$ is rounded up, "N2".

| PP | RM | CP | Sep | N0 | N1 | N2 |
|---|---|---|---|---|---|---|
| 10.84 | 8.14 | 73.49 | 4.14 | 5.87 | 16.17 | 77.91 |

Table 4: Average time partition (in %)

# 7 A short review of heuristics based on column generation

Several classical heuristics can be adapted to the context of a column generation approach to obtain primal bounds. Let us briefly review previous work where greedy, local search, rounding or other LP based heuristics have been used in a decomposition approach.

**Solving the Restricted Master Integer Program (RMIP):** A natural way to obtain an integer solution is to solve the master restricted to the set of generated columns as an integer program (by Branch-and-Bound). One can set a limit on the time and number of nodes allowed by the solver. Then, a depth-first-search might be the best strategy. Implementations of this approach vary mainly by the definition of the restricted column set $\overline{Q}$: $\overline{Q}$ can contain some or all the columns generated during the LP optimization procedure [18, 17, 16] or columns generated during other heuristic algorithms [47, 46].

**Implementing a Greedy Heuristic (GH):** A greedy heuristic for the set covering type formulation defined by the master program consists in iteratively adding a greedy selected column in the partial solution until feasibility is reached. Implementation variants concern the criteria for selecting the column to add to the partial solution. It can be based on the column costs, the ratio of their costs to their contribution to the set covering constraints, or their reduced cost. The latter can be computed exactly or approximately; using dual price estimates obtained from a dual heuristic or LP optimal dual prices. One can restrict its attention to columns that are "proper" (see [51]) for the residual problem. In [43], the column selection criterion is the reduced cost computed with LP optimal dual values, while in [9] dual prices are updated after fixing a column. In [16, 24] columns are selected based on "pseudo-costs".

**Rounding Heuristic based on the LP master solution (RH):** A rounding heuristic is a heuristic search plunging depth into the branch-and-price tree: each branch consists in selecting a master column with fractional value and rounding it up; at each node, the residual master program is re-optimized. Generating new columns in the process of optimizing the linear relaxation of the residual problem is an important feature for the success of the approach as it allows to construct feasible solutions (while the RMIP approach often fails to obtain feasible primal solutions). It is indeed to be noticed that the solution space of the residual problem can be quite different from that of the original problem and hence columns that were generated at the root might quickly become obsolete. The

greedy heuristic approach can be seen as a special case where only one column is generated in the re-optimization. Observe that rounding a fractional column downwards is not an option because the pricing problem should then be amended to avoid regenerating this specific column. However, a variant can be to round variables of the original formulation instead of that of the Dantzig-Wolfe reformulation ([28, 36] branch on variables of the pricing problem).

The literature reports on several other variants of the RH approach. The root node master LP can be solved at optimality or not (as in [33]). The re-optimization after branching can also be truncated (see [11, 49]). The selection of the column(s) to be rounded is crucial: it can be chosen according to $LP$ value [33, 49], or using a greedy score that can combined LP information with greedy arguments [11]. One can fix one column at a time [43, 49] or several [9, 33, 36, 39]. The RH can be combined with the two above approaches: the RH can be stopped early and a RMIP approach or a greedy heuristic can be applied to the residual problem (as in [9, 39]). Partial backtracking can allow to diversify the search defined by the deterministic plunge into the branch-and-price tree ([49] explores different branches hanging from the root; [9, 28, 33, 39] change the value of some variables and re-apply the RH), or, if some randomness is used for selecting the rounded variables, several passes of the RH procedure might be applied. Finally, the RH heuristic can be applied before reaching LP optimality [28] or between each cut generation phase [9], or from different branch-and-price tree nodes.

**Local Search Heuristics (LS):** In the context of a column generation approach, local search consists in exploring the neighborhood of the current master solution, $\hat{\lambda}$, in search for a better solution and to repeat this procedure. To the current solution, $\hat{\lambda}$, one associates a set of columns, $Q(\hat{\lambda})$, that is typically defined as the support of $\hat{\lambda}$ plus some extra "good" columns. A neighbor solution is defined by modifying column set $Q(\hat{\lambda})$ (by removing and adding columns) and extracting a new solution from the newly defined restricted master. Meta-heuristics can then be implemented from this LS paradigm (f.i., [24] proposes a simulated annealing approach to a cutting stock problem where the solution extracted from a column set $\overline{Q}$ is defined by solving heuristically the restricted master integer program and a neighbor solution is obtained by removing and adding columns to $\overline{Q}$; a similar approach is used in [5]). Column subset can be extracted from a pool, or new columns might be generated in the process of building a solution from a given column subset. Other references to LS in a column generation context have a different aim: [18] develops collaborative approaches between column generation and

LS where a local search heuristic is made less myopic using LP information derived from a column generation solution procedure; [29] uses column generation to optimize the choice of a neighbor in a large neighborhood.

# 8 Primal solutions to the tactical planning problem

We now outline how the above methods, denoted RMIP, GH, RH, and LS, can be implemented for our application and we comment on the performance of our implementation. In all cases, the primal solutions are defined for the discrete master formulation. However, any LP information (primal or dual) required by the heuristics is obtained by solving the aggregate master program. When a partial solution is recorded or updated, master and pricing problems must be updated too. In the discrete master, the right-hand-side of customer demand covering constraints is set to zero for pairs $(i, t)$ already collected in the partial solution and a fixed number of vehicles is recorded as already used in the vehicle upper bound constraints for some periods. In the aggregate master, the fractions of demand remaining to be covered are adjusted, while the average vehicle use is updated. We pass on to the pricing problem, restrictions that are specific to some starting dates (new routes must not cover demands that are already covered, or cannot use a vehicle in periods where no more vehicles are available). Hence, even when dealing with the aggregate master, one must iterate on different core pricing sub-problems for each starting dates.

To implement the RMIP heuristic approach, we define the restricted set of columns as all the columns generated during the LP solution of the aggregate master which we duplicate for each feasible starting dates. We formulate the restricted discrete master program using the discrete variables: $\lambda_{rs} = 1$ if the aggregate column $r$ is taken with the starting date $s$. Our computational experiments show that solving the restricted discrete master program to integer optimality is quite computationally intensive. Indeed, for instances with 100 customers, we have found no solution after 2 hours of computing time. This is partially due to the inherent symmetry in the discrete master program. Hence, we abandon our hope to generate primal solutions in this way.

In our implementation of the GH approach, we generate an aggregate column compatible with the partial solution to the discrete master by solving the pricing problem at optimality. This column is fixed to value one and a starting date is randomly chosen among the feasible possibilities. We call this heuristic twice: once at the outset of the algorithm, using our dual heuristic to obtain dual price estimates (30); and a second time,

at the end of the LP optimization of the aggregate master, using the LP optimum dual prices. Computational tests on the instances of Section 6 are reported in Table 5. The columns gives the instance class name, the gaps to the best dual bounds of Section 6, the total time, and the percentage of this time spent in the greedy heuristic. GH is quite fast but the optimality gap are important. Indeed, we observed that extra columns must typically be selected to complete the last holes in the planning and this induces an increase in the number of vehicles that are used.

| Name | gap | Total time | % of time spent in GH |
|---|---|---|---|
| av RAND100 | 55.23 | 1m32s | 8 |
| av IND60 | 36.50 | 33s | 8 |
| IND172 | 64.28 | 8m45s | 8 |
| IND157 | 40.04 | 3m30s | 12 |
| **av 17 inst** | **49.36** | | |

Table 5: Results for the Greedy Heuristic

The implementation of the RH is more complex. We compared several implementations varying on the criteria for selecting a column from the aggregate master LP solution. We also tested different ways of fixing the starting date of the selected columns. We varied the effort put in re-optimizing the aggregate master LP associated to the residual problem. We tested diversification strategies through partial backtracking. And, we tried calling the RH at different stages. Observe that, in the process of the RH, one might encounter an integer solution to the aggregate master (this often happens after fixing a large part of the variables). Then, it remains to see whether there exists an associated integer solution to the discrete master residual problem. To this end, we solve to IP optimality the later discrete master restricted to the current pool of columns. The fact that a partial solutions has already been fixed not only reduces the size of the residual problem but also breaks the symmetry. Hence, these restricted discrete masters can be solved quickly in practice (contrary to our experience with the RMIP approach).

In Table 6, we compare several variants of the RH on the test problems of Section 6. The variant named "Basic" consists in calling RH on the LP optimal solution to the aggregate master at the root node. The number of iterations in re-optimizing the master LP is bounded by 300. We diversify the search by calling the RH 3 times, ensuring that the first column being rounded-up is different on each 3 passes. The rounded column and its starting date are both chosen using a deterministic criterion: we select the column $r$ and the starting date $s$ with the smallest score based on the ratio of column cost per

unit of constraint satisfaction. The column cost is equal to $\delta_{rs} + \alpha\frac{c^r}{p^r}$ where $\delta_{rs} = 1$ if one must use an additional vehicle after this fixation. The constraint satisfaction measure considers constraints (24) and (25). It is defined as $\frac{\sum_{i,\ell} \ell x_{i\ell}}{p^r} \times p^r = \sum_{i,\ell} \ell x_{i\ell}$. This basic variant is compared to others. The implementation named "Random" consists in selecting the rounded-up column, $r$, randomly, with a probability proportional to $min(\lambda_r, 1)$ and fixing its starting date randomly. Variant named "100cg" is the basic variant where the aggregate master LP re-optimization is limited to 100 iterations of the column generation procedure. Variant named "every200" consists in applying the basic variant every 200 iterations of the column generation procedure (if there is no artificial columns in the $LP$ solution). Variant named "cut" consists in applying the RH after the cutting plane procedure (on a solution that is very fractional).

Table 6 reports optimality gaps, number of vehicles used in the primal solution, $V$, and total computing times. RH provides better solutions than GH. The fastest variant is obviously "100cg". Note that the computing time for "cut" is large due to the difficulty in re-optimizing the master after adding cuts. With "Random", we have no solution for IND157. The variant with the best optimality gap and a minimal number of vehicles is "Basic". In Table 7, we report even better results obtained by calling "Basic" at root node and after branching on $Vaver$: the average gap is 11.66%. For all instances but one, we can show that the number of vehicles used in our primal solution is minimum because we have branched on $Vaver$ and Node 1 has been proved infeasible. By imposing some restrictions on the solution space, such as further restricting the set $P = \{1, 2, 3\}$, we sometime get even smaller optimality gaps (we then obtain a gap of 9,67% on average).

| Name | av RAND100 | | | av IND60 | | | IND172 | | | IND157 | | | **av 17 inst** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | gap | V | Time | gap | V | Time | gap | V | Time | gap | V | Time | gap | V |
| Basic | 14.12 | 4 | 10m5s | 18.39 | 3.2 | 3m33s | 6.89 | 6 | 37m56s | 10.86 | 7 | 42m3s | **14.76** | **4.05** |
| Random | 41.43 | 5.5 | 10m1s | 38.31 | 4 | 3m50s | 17.83 | 7 | 1h12m | NO | | 14m42s | | |
| 100cg | 15.04 | 4 | 4m41s | 27.2 | 3.4 | 1m47s | 8.993 | 6 | 20m58s | 14.77 | 7 | 9m23s | 21.18 | 4.11 |
| every200 | 14.30 | 4.1 | 19m57s | 25.88 | 3.4 | 5m57s | 6.896 | 6 | 37m56s | 6.586 | 7 | 1h45m49s | 16.82 | 4.18 |
| cut | 32.55 | 4.9 | 46m23s | 19.86 | 3.2 | 14m46s7 | 7.041 | 6 | 1h54m32s | 25.85 | 8 | 2h25m23s | 26.92 | 4.65 |

Table 6: Comparing variants of the Rounding Heuristic.

For LS, we obtain a neighbor solution by removing a few columns from the current integer solution and re-building a complete integer solution with the above basic rounding heuristic procedure. The choice of columns temporarily deleted from the pool is important. We test two variants: either a random selection of 6 (= $Pmax$) columns are deleted

| Name | gap | V | Total time | % of time spent in RH |
|---|---|---|---|---|
| av RAND100 | 12.29 | 4 | 18m22s | 90 |
| av IND60 | 11.75 | 3 | 6m10s | 78 |
| IND172 | 5.574 | 6 | 1h22m2s | 79 |
| IND157 | 10.86 | 7 | 1h30m16s | 91 |
| **av 17 inst** | **11.66** | **4** | | |

Table 7: Basic Rounding Heuristic called at the root and after branching.

from the primal solution, or we make a deterministic selection as follows. We delete columns from the primal solution that seem to be "poor". We arbitrarily assess a column as poor if its load is less than $\frac{1}{3}W$. If none qualifies as poor, we pick a column at random. This first selection of columns is denoted by set $Q^0$. Then, we select as candidate for deletion along $Q^0$, other columns that either concern the same periods (a set denoted $Q^1$) or share the same vehicle (a set denoted $Q^2$). The purpose is to favor the exchanges of customers between clusters used in the same periods and to attempt to free a vehicle. $Q^1$ is the set of columns from the primal solution that have the same periodicity and starting date than a column in $Q^0$. $Q^2$ is the set of columns from the primal solution that have the same periodicity than a column in $Q^0$ and result in a decrease in the number of vehicle used when deleted from the solution. $Q^0 \cup Q^1 \cup Q^2$ defines the set of candidate columns for deletion. From $Q^1$, we pick at most 1 column for each pair of periodicity and starting date, that with the minimum score $\frac{dist^r}{W-load^r} * (1 + \frac{nbDel^r}{2})$, where $dist^r$ represents the distance between $r$ and the closest deleted columns from $Q^0$ (i.e., the distance between customers in the two columns), $load^r$ is the load and $nbDel^r$ is the number of times that column has been deleted in past trial. Similarly, we pick at most one column from $Q^2$ for each periodicity $p$ and starting date $s$. Rebuilding a solution with the RH is computationally expensive. Hence, we restrict the number of trial in our exploration of the neighborhood to at most 10. In rebuilding a solution with the RH, the first rounded-up column is not selected among columns that have just been deleted or that have already been selected as first column in previous trial.

In Table 8, we compare these two variants in trying to improve the first solution found by the RH, i.e., the test is limited to a single application of the LS algorithm starting from the first primal solution that we encounter, with at most 10 trial in exploring the neighborhood of a solution. Table 8 reports the improvement, "impr", (expressed as a percentage) that was obtained, the number of times we succeed in finding an improving solution, "succ", the average number of trial needed in our exploration of a neighborhood before finding an improving solution (if we do find one; it is bounded by 10), "trial",

and the total computing time. We always find some improving solution when using the deterministic variant. The improvements vary from 1 to 15%. In Table 9, we present the results using the deterministic LS on all solutions obtained during all the passes of the basic RH at the root node. The overall computational effort is however bounded by restricting to 500 the number of columns that can be generated during the primal heuristic. The percentage of time spent in LS includes the time to rebuild a primal solution using the RH approach. The comparison of GH, RH and LS is summarized in Table 10. It shows that the optimality gap decreases from 14.76 to 11.22 thanks to LS. Moreover, we obtain the minimal numbers of used vehicles.

| | Random selection of deleted columns | | | | Deterministic selection of deleted columns | | | |
|---|---|---|---|---|---|---|---|---|
| Name | % impr in primal bd | succ | trial | time | % impr in primal bd | succ | trial | time |
| av RAND100 | 1.60 | 7/10 | 5.1 | 6m33s | 8.43 | 10/10 | 1.9 | 4m47s |
| av IND60 | 4.25 | 3/5 | 5 | 3m11s | 8.27 | 5/5 | 2.4 | 1m58s |
| IND172 | 0.00 | 0/1 | - | 25m47s | 0.34 | 1/1 | 6 | 21m28s |
| IND157 | 0.34 | 1/1 | 2 | 27m17s | 0.26 | 1/1 | 1 | 26m43s |
| **av 17 inst** | 2.11 | 11/17 | 4.4 | | **7.43** | 17/17 | **2.2** | |

Table 8: Comparison of 2 LS variants applied to the first primal solution generated with RH.

| Name | gap | V | Total time | % of time spent in RH | % of time spent in LS |
|---|---|---|---|---|---|
| av RAND100 | 11.88 | 4 | 46m4s1 | 19 | 76 |
| av IND60 | 11.74 | 3 | 13m27s | 22 | 74 |
| IND172 | 5.735 | 6 | 1h0m50s | 48 | 38 |
| IND157 | 7.389 | 7 | 54m50s | 71 | 24 |
| **av 17 inst** | **11.22** | **4** | | | |

Table 9: Results with the deterministic LS applied on each primal solutions generated during the 3 passes of the basic RH.

| Method | average gap |
|---|---|
| GH | 49.36 |
| RH at the root node only | 14.76 |
| RH through the 3 b-a-p nodes | 11.66 |
| LS at the root node only | 11.22 |

Table 10: Comparison of average results with GH, RH and LS approaches on the test problems of Section 6.

# 9 Results for large industrial instances

Finally, we applied our approach to a large real-life industrial instance with 260 customers. It is quite difficult to collect real-life data from industry. This test problem represents the on-the-ground situation for a restricted geographical area during year 2005 (we may not say more due to our confidentiality agreement). At the planning level, a period correspond to a week. More than 60% of the customers have a maximum time lag between visits that is inferior to 6 weeks (our maximum periodicity). The others have a $t^{\max}$ ranging from 7 to 14 weeks. Applying partial branching and cutting planes to compute the best possible dual bound requires 8h34m of computing time. The "DB+br+cut" dual bound has value 838.17. The cutting plane procedure alone requires 5h21m.

In a different run, skipping the time consuming cutting plane procedure, we apply our primal heuristics. Applying the basic RH combined with partial branching gives a primal solution with a gap of approximately 9% from the above dual bound in 4h29m (this run does not call on the cutting plan procedure; moreover, Node 1 has been proved infeasible in our dual bound computation; therefore, it is not explored anymore). The number of vehicles is equal to 9 (it is optimal as proved by the fact that the branch with $Vaver \leq 8$ is infeasible). In another trial, we restrict ourself to the root node and apply the basic RH followed by LS. We again obtain a solution with a gap of approximately 9% from the above dual bound and a number of vehicles equal to 9 in 3h40m.

We obtain even better results by restricting the search space to solution using periodicities in the set $P = \{1, 2, 3\}$. Applying the RH plus partial branching or RH and LS at the root node followed by a post-optimization procedure, we obtain a primal solution with a gap of 6% in 1h49m and 2h53m respectively. The post-optimization procedure consists in transforming one route of periodicity 3 into two routes of periodicity 6 while this improves the solution (it can allow to visit a customer every 6 periods instead of 3). The feasible splitting scenarios are built into a integer program whose solution provides a selection minimizing routing costs. The best solution is pictured in Figure 1 for a specific period. Six clusters (each of which is associated to a specific color) are drawn to show their form resulting from the cost structure. They include some sites that are on the path to the garage G or to the depot D. But the majority of the sites are well gathered around the center. The results are summarized in Table 11.

In Table 12, we compare this solution to the one used by our industrial partner on an

| Algorithm | gap (in %) | V | total time |
|---|---|---|---|
| RH + partial branching with $P = \{1, 2, 3, 4, 5, 6\}$ | 9.25 | 9 | 4h29m |
| RH + LS at the root node with $P = \{1, 2, 3, 4, 5, 6\}$ | 8.92 | 9 | 3h40m |
| RH + partial branching with $P = \{1, 2, 3\}$ + post-optim | 6.67 | 9 | 1h49m |
| RH + LS at the root node with $P = \{1, 2, 3\}$ + post-optim | 6.23 | 9 | 2h53m |

Table 11: Comparison of average results with RH and LS approaches on the industrial test case of year 2005 (with 260 customers).

| | av # of customer visits per week | V | av travel distance per week |
|---|---|---|---|
| sol of the tactical planning model | 98.5 | 9 | 711 km |
| industrial solution | 59 (*) | 10 | 782 km |

Table 12: Comparison of our best primal solution to the tactical planning industrial problem of year 2005 with the industrial solution – (*) In the industrial solution, a visit is not counted if, once on site, the driver decides not to collect the customer stock because he finds it too low.

average week of year 2005. From what we know, this industrial solution was built on a day-to-day basis looking at the customer requiring the most urgent service and trying to achieve some regional clusterization. We are given the total number of vehicles used in each period, the number of collects and the travel distance in the operational solution used by the industry in 2005. We estimate the average behavior on a week by dividing these numbers by 52. For our tactical solution, we compute the travel distance by heuristically solving a TSP for each cluster. We observe that our solution requires 9 vehicles instead of 10 in the averaged-out industrial solution. The travel distance of our solution is 10 percent inferior to that of the industrial solution. We collect more sites than in the industrial solution (i.e., we tend to collect a site before it reaches its maximum filling capacity). Indeed, visiting a site that is on a vehicle route costs 5 minutes of collect time, while having to do a detour later to visit it, can cost much more, since travel times between sites are typically larger than 5 minutes. Moreover, visiting sites on a more regular basis can be an edge against uncertainty and helps to avoid stock-out situation. We believe that the regional clusterization taken into account in our tactical planning model brings a plus that makes the difference. The next step would be to build operational solutions using our tactical solution as a target.

# Conclusion

The scope of the paper is twofold. We solve a large scale industrial inventory routing problem untreated in the literature. Moreover, we show that a mathematical program-

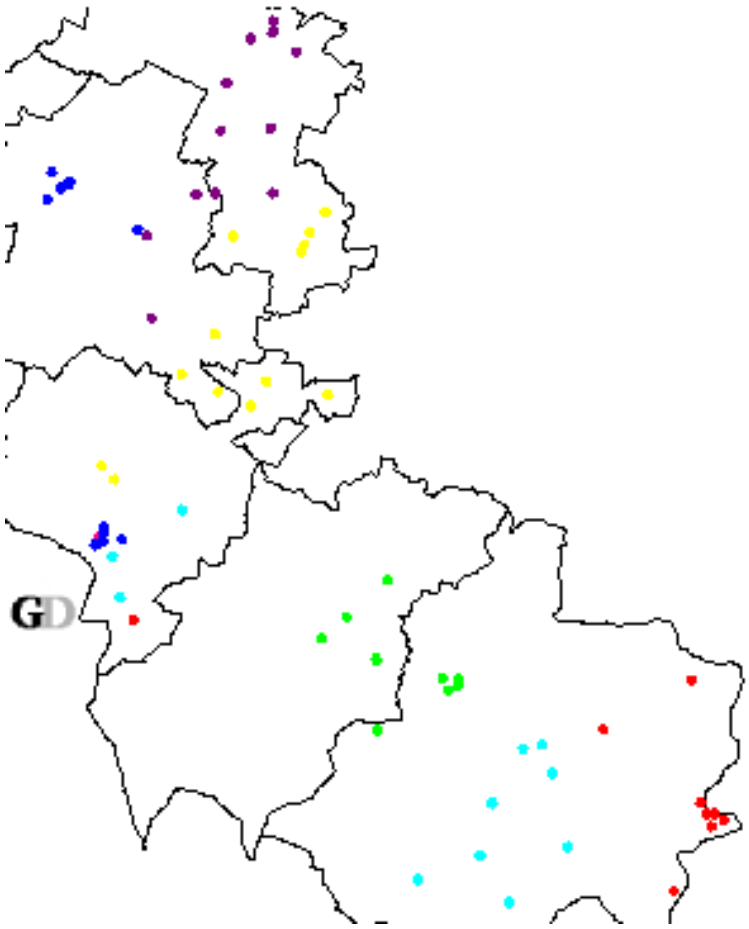

Figure 1: Sample of a solution on industrial data

ming based approach (more precisely a truncated Branch-and-Price-and-Cut algorithm combined with primal heuristics) allows to obtain solutions for large scale problems and to compute a bound on the deviation to optimality. Our preliminary comparison with the industrial solution gives hope that our work might have a practical impact on the industrial practice. A key to the success of our approach is a formulation modeling an average behavior to avoid symmetry.

## Acknowledgments

We are very grateful to our industrial partner, F. Rodes, for sharing with us problem description and real-life data. We acknowledge his financial support that helped in conducting this study. We own a special thank to P. Pesneau who refereed our initial draft.

# References


[1] EH. Aghezzaf, B. Raa, and H. Van Landeghem. Modeling inventory routing problems in supply chains of high consumption products. *European Journal of Operational Research*, 169:1048–1063, 2006.

[2] F. Al-Khayyal and SJ. Hwang. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part i: Applications and model. *European Journal of Operational Research*, 176:106–130, 2007.

[3] S. Anily and A. Federgruen. One warehouse multiple retailer systems with vehicle routing costs. *Management Science*, 36:792–114, 1990.

[4] C. Archetti, L. Bertazzi, G. Laporte, and MG. Speranza. A branch-and-cut algorithm for a vendor managed inventory routing problem. *Transportation Science*, 2007. to appear.

[5] C. Archetti, MWP. Savelsbergh, and MG. Speranza. An optimization-based heuristic for the split delivery routing problem. Technical report, Department of Quantitative Methods, University of Brescia, 2006.

[6] P. Avella, M. Boccia, and A. Sforza. Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research*, 152:170–179, 2004.

[7] JF. Bard, L. Huang, P. Jaillet, and M. Dror. A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science*, 32:189–203, 1998.

[8] W. Bell, L. Dalberto, M. Fisher, A. Greenfield, R. Jaikumar, P. Kedia, R. Mack, and P. Prutzman. Improving the distribution of industrial gases with on-line computerized routing and scheduling optimizer. *Interfaces*, 13:4–23, 1983.

[9] G. Belov and G. Scheithauer. A cutting plane algorithm for the one dimensional cutting stock problem with multiple stock lengths. *European Journal of Operational Research*, 141:274–294, 2002.

[10] L. Bertazzi, G. Paletta, and MG. Speranza. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36:119–132, 2002.

[11] R. Borndörfer, M. Grötschel, and A. Löbel. Scheduling duties by adaptive column generation. Technical report, ZIB-Report, 2001.

[12] J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems. *Operations Research*, 43:649–660, 1995.

[13] O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical Programming*, 2006.

[14] AM. Campbell, LW Clarke, and MWP. Savelsbergh. *Inventory routing in practice*, chapter 12, pages 309 – 330. SIAM Monographs on Discrete Mathematics and Applications, 2002.

[15] AM. Campbell and MWP. Savelsbergh. A decomposition approach for the inventory routing problem. *Transportation Science*, 38:488–502, 2004.

[16] A. Ceselli, G. Righini, and M. Salani. A column generation algorithm for a vehicle routing problem with economies of scale and additional constraints. In *Sixth Triennial Symposium on Transportation Analysis*, Phuket, Thailand, 2007.

[17] A. Chabrier. Heuristic branch-and-price-and-cut to solve a network design problem. In *Proceedings CPAIOR*, Montréal, Canada, mai 2003.

[18] A. Chabrier, E. Danna, and C. Le Pape. Coopération entre génération de colonnes et recherche locale appliquées au problème de routage de véhicules. In *Huitièmes Journées Nationales sur la résolution de Problèmes NP-Complets (JNPC)*, pages 83–97, Nice, France, mai 2002.

[19] LMA. Chan, A. Federgruen, and D. Simchi-Levi. Probabilistic analyses and pratical algorithms for inventory-routing models. *Operations Research*, 46:96–106, 1998.

[20] TW. Chien, A. Balakrishnan, and RT. Wong. An integrated inventory allocation and vehicle routing problem. *Transportation Science*, 23:67–76, 1989.

[21] M. Christiansen. Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science*, 33:3–13, 1999.

[22] M. Christiansen and B. Nygreen. A method for solving ship routing problems with inventory constraints. *Annals of Operations Research*, 81:357–378, 1998.

[23] N. Christofides and S. Eilon. Expected distances in distribution problems. *Operational Research Quaterly*, 20:437–443, 1969.

[24] FL. Cimelière. *Optimisation du traitement de l'ordre de fabrication dans l'industrie textile*. PhD thesis, Université Bordeaux 1, France, 2004.

[25] JF. Cordeau, G. Laporte, MWP. Savelsbergh, and D. Vigo. *Vehicle Routing*, volume 14, pages 367 – 428. Handbooks in OR and Management Science, Elsevier, 2007.

[26] K. Cousineau-Ouimet. A tabu search heuristic for the inventory routing problem. In *Proceedings of the 37th Annual ORSNZ conference*, Auckland, New Zealand, novembre 2002.

[27] AL. Custódio and RC. Oliveira. A system of logistic management integrating inventory management and routing. In *Optimization 2001*, Aveiro, Portugal, juillet 2001.

[28] Z. Degraeve and R. Jans. A new dantzig-wolfe reformulation and branch-and-price algorithm for the capacitated lot sizing problem with set-up times. *ERIM Report Series in Management*, ERS-2003-010-LIS, 2003.

[29] G. Desaulniers, E. Prescott-Gagnon, and LM. Rousseau. A large neighborhood search algorithm for the vehicle routing problem with time windows. In *Sixth Triennial Symposium on Transportation Analysis*, Phuket, Thailand, 2007.

[30] M. Dror and M. Ball. Inventory/routing: reduction from an annual to a short-term problem. *Naval Research Logistics*, 34:891–905, 1987.

[31] A. Federgruen and D. Simchi-Levi. *Analysis of Vehicle Routing and Inventory Management Problems*, volume 8 of *Handbooks in OR and Management Science*, pages 297–371. Amsterdam, Elsevier edition, 1995.

[32] ML. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11:109–124, 1981.

[33] M. Gamache, F. Soumis, G. Marquis, and J. Desrosiers. A column generation approach for large scale aircrew rostering problem. *Operations Research*, 47:247–263, 1999.

[34] V. Gaur and ML. Fisher. A periodic inventory routing problem at a supermarket chain. *Operations Research*, 52:813–822, 2004.

[35] B. Golden, A. Assad, and R. Dahl. Analysis of a large scale vehicle routing problem with an inventory component. *Large Scale System 7*, pages 181–190, 1984.

[36] O. Günlük, T. Kimbel, L. Ladangi, B. Schieber, and G. Sorkin. Vehicle routing and staffing for sedan service. Technical Report RC23208, IBM Research Report, 2005.

[37] IRP instances. http://www.math.u-bordeaux.fr/~smichel/irpinstances.tar.

[38] P. Jaillet, J. Bard, L. Huang, and M. Dror. Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science*, 36:292–300, 2002.

[39] K. Kiwiel. An inexact bundle approach to cutting stock problems. Technical report, 2005.

[40] AJ. Kleywegt, V. Nori, and MWL. Savelsbergh. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science*, 38(42-70), 2004.

[41] V. Malépart, F. Boctor, J. Renaud, and S. Labillois. Nouvelles approches pour l'approvisionnement des stations d'essence. *Revue Française de Gestion Industrielle*, 22:15–31, 2002.

[42] GL. Nemhauser and LA. Wolsey. *Integer and Combinatorial Optimization*, chapter II.1.4 : *The theory of valid inequalities: superadditive functions and valid inequalities*, pages 229–237. John Wiley & Sons, 1988.

[43] N. Perrot. *Integer Programming Column Generation Strategies for the Cutting Stock Problem and its Variants*. PhD thesis, Université Bordeaux 1, France, 2005.

[44] D. Pisinger. A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research*, 83:394–410, 1995.

[45] W. Qu, J. Bookbinder, and P. Iyogun. An integrated inventory transportation system with modified periodic policy for multiple products. *European Journal of Operational Research*, 115:254–269, 1999.

[46] V. Schmid, KF. Doerner, RF. Hartl, MWP. Savelsbergh, and W. Stoecher. An effective heuristic for ready mixed concrete delivery. In *Sixth Triennial Symposium on Transportation Analysis*, Phuket, Thailand, 2007.

[47] E. Taillard. A heuristic column generation method for the heterogeneous vrp. *RAIRO Operations Research*, 33:1–14, 1999.

[48] D. Taqa Allah, J. Renaud, and F. Boctor. Le problème d'approvisionnement des stations d'essences, the gas stations supply problem. *Journal européen des systèmes automatisés*, 34:11–33, 2000.

[49] F. Vanderbeck. Exact algorithm for minimising the number of setups in the one-dimensional cutting stock problem. *Operations Research*, 48:915–926, 2000.

[50] F. Vanderbeck. *Implementing Mixed Integer Column Generation*, pages 331–358. Springer, 2005.

[51] F. Vanderbeck and MPW. Savelsbergh. A generic view of dantzig-wolfe decomposition in mixed integer programming. *Operations Research Letters*, 34:296–306, 2006.

[52] I. Webb and R. Larson. Period and phase of customer replenishment: A new approach to the strategic inventory/routing problem. *European Journal of Operational Research*, 85:132–148, 1995.

[53] M. Witucki, P. Dejax, and M. Haouari. Un modèle et un algorithme de résolution exacte pour le problème de tournées de véhicules multipériodique: une application à la distribution des gaz industriels. In *Deuxième congrès international franco-québecois de génie industriel*, Albi, France, 1997.