# Inventory Routing Problem solved by Heuristic based on Column Generation

S. Michel F. Vanderbeck

Applied Mathematics, University Bordeaux 1,
351, Cours de la Liberation 33405 Talence Cedex, FRANCE
*smichel/fv@math.u-bordeaux1.fr*

We consider an application of the inventory routing 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. On each visit, the vehicle empties the stock of accumulated product. In the tactical planning phase, we search for a periodic solution to be repeated on an infinite time horizon. The objectives are to minimise the vehicle fleet size as well as the transportation cost, while achieving some form of regional clusterisation in partitioning the sites between the vehicles. The structure of the problem is exploited to develop a Dantzig-Wolfe decomposition approach. The column generation subproblem yields periodic routes. It reduces to a multiple choice knapsack problem. The issues related to the construction of the planning are dealt with in a master program. The latter program is reformulated in terms of aggregated variables to avoid the symmetry in time. Cutting planes are added to improve the formulation. Dual bounds are obtained by LP relaxation and tightened through partial branching. Primal bounds are derived through rounding and local search procedures specially developed for use in the context of a column generation approach. Real-life instances of the problem are solved with reasonable optimality gaps.

# 1 The Problem

The Inventory Routing Problem (IRP) combines issue of routing for pick-up or deliveries with inventory management at customer sites. Three decisions have to be made: (i) when to serve a customer; (ii) how much to deliver to a customer when it is served; and (iii) which delivery routes to use. Many variants are discussed in the literature [1, 2, 3].

The application considered here concerned the design of routes for collecting a single product from customers who accumulate it in their stock. At the tactical planning level, filling rates at collection points are seen as determinist. The stock management rule is simple: at each pick-up, the stock is emptied (this is the equivalent of 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, $tmax$. The stock management costs reduce to the transportation cost.

In search for a periodic solution, we restrict the solution space by imposing that route periodicity are selected from a restricted set $P$: for example $P = \{1, 2, 3, 4, 6\}$. 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 least common multiple of the periodicities (in our example $H = 12$). $H$ is the length of the regeneration cycle. The planning requirements boil down to ensuring that the stocks produced on each period of the regeneration cycle are picked-up in some route.

The exact routing of vehicles is considered as an operational issue. At the tactical planning level, we define a route by the cluster of visited customer sites. The quantities collected at each visit point are defined in such a way that their sum does not exceed the vehicle capacity. The operational routing cost is approximated by the sum of the distances to the cluster center defined as one of the visited points (it is the seed of the route). Thus, each route is associated to a star in the graph of customer points. This measure favors the grouping of customers that are geographically close to each other. The planning constraints will induces the formation of clusters that group customers sharing the same frequency of collect.

The cost function includes fixed costs per vehicle (the main objective being to minimize the number of vehicle used), but it also includes our cluster approximation of routing costs.

## 2 A Dantzig-Wolfe Decomposition approach

The problem decomposes into planning issues on the hand and routing issues on the other. We formulate the planning problem in terms of variables associated with the selection of routes [5]. The definition of a route includes the visited customer, the quantities picked-up at each site (expressed in number of periods worth), the periodicity and the 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: let $\phi_{i\ell}$ equal to 1 if the customer $i$ is in the cluster and is collected for $\ell$ periods; the associated profit is $p_{i\ell}$; the knapsack formulation is

$$\max \quad \textstyle\sum_{i\ell} \quad p_{i\ell}\phi_{i\ell} \tag{1}$$

$$\sum_{\ell} \phi_{k\ell} \quad = \quad 1 \tag{2}$$

$$\sum_{\ell} \phi_{i\ell} \quad \leq \quad 1 \qquad \forall\, i \neq k \tag{3}$$

$$\sum_{i\ell} \ell d_i \phi_{i\ell} \quad \leq \quad W \tag{4}$$

$$\phi_{i\ell} \quad \in \quad \{0,1\} \qquad \forall\, i, \ell \tag{5}$$

where $d_i$ is the accumulation rate at customer site $i$ and $W$ is the vehicle capacity.

Let $\{(c^q, \phi^q, p^q, s^q)\}_{q \in Q}$ be the enumerated set of periodic routes, $q$, are defined as the solution, $\phi^q$ to the above knapsack subproblem along with their cost, $c^q$, and the definition of a periodicity $p^q$ and a starting date, $s^q$. From the information given by $(\phi^q, p^q, s^q)$, one can generate an indicator matrix $\delta^q$ with $\delta^q_{it} = 1$ if the demand of period $t$ for customer $i \in N$ is covered by route $q$ and zero otherwise, while $\delta^q_{0\,t} = 1$ if the vehicle is used in period $t$ and zero otherwise. Then, the inventory routing problem can be formulated as:

$$Z^d_{IP} = \min\ Vmax \quad + \quad \alpha \sum_{q \in Q} \frac{c^q}{p^q}\, \lambda_q \tag{6}$$

$$\sum_{q} \delta^q_{it} \lambda_q \quad \geq \quad 1 \qquad \forall i = 1, \ldots, n;\ t = 1, \ldots, H \tag{7}$$

$$\sum_{q} \delta^q_{0t} \lambda_q \quad \leq \quad Vmax \qquad \forall t = 1, \ldots, H \tag{8}$$

$$\lambda_q \in \{0,1\} \qquad \forall\, q \tag{9}$$

$$Vmax \in I\!N. \tag{10}$$

where $0 \leq \alpha < 1$ is a coefficient to balance both term in the objective, $\lambda_q = 1$ if periodic route $q$ is used and zero otherwise, while $Vmax$ is the maximum number of vehicles used in a period. The variables $\lambda_q$ and associated columns are generated dynamically in the course of the optimisation procedure (using a column generation approach).

The above formulation suffers from a symmetry in $t$: equivalent solutions can be defined that differ only by a permutation in the choice of starting dates. To avoid this drawback, we aggregate periods and model an average behavior. Technically speaking, we implement a state space relaxation in the space of columns: aggregating all columns that differ only by their starting dates, we project our column space as follows

$$\{(c^q, \phi^q, p^q, s^q)\}_{q\in Q} \quad \underrightarrow{proj} \quad \{(c^r, \phi^r, p^r)\}_{r\in R}.$$

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:

$$Z^a_{IP} = \min Vaver \quad + \quad \alpha \sum_{r\in R} \frac{c^r}{p^r}\, \lambda_r \tag{11}$$

$$\sum_{r\in R} \frac{\ell}{p^r} \phi^r_{i\ell} \lambda_r \quad \geq \quad 1 \qquad \forall i \tag{12}$$

$$\sum_{r\in R} \frac{1}{p^r} \lambda_r \quad \leq \quad Vaver \tag{13}$$

$$\lambda_r \in I\!N \qquad \forall\, r \tag{14}$$

$$Vaver \in I\!N. \tag{15}$$

where $\lambda_r$ is the number of times that a vehicle uses periodic route $r$ and $Vaver$ is the average number of vehicles used per period.

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. 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.

# 3 Branch-and-Price-and-Cut

To obtain dual bounds, the aggregate master is solved to LP optimality by column generation (the pricing problem (1-5) is solved by dynamic program [4]). Then, a cutting planes procedure is implemented base on a family of valid inequalities that we derived from (12) using a MIR procedure. Let $h$ be an integer ranging from 1 to $H-1$ and $i \in N$ such that $tmax_i > 1$, theses inequalities take the form:

$$\sum_{r,h\ell\%p=0} \frac{\ell}{p}\phi^r_{i\ell}\lambda_r + \sum_{r,h\ell\%p\neq 0} (\lceil \frac{h\ell}{p} \rceil - \frac{h\ell}{p})\phi^r_{i\ell}\lambda_r \geq 1 \tag{16}$$

After each addition of a cut, we return to the column generation procedure. Then, we further improve the dual bound through a truncated branch-and-bound procedure: we branch on the $Vaver$ variable only. Given the structure of our objective that focuses on vehicle use, this branching has an important impact of the bound. Moreover, the branch were $Vaver$ is rounded down can often be proved infeasible. The dual bound improvement observed by adding cut is small (less than 2% in our numerical tests), but the improvement obtained through partial branching can get bigger (depending on the instance, it ranges from less than 1% up to more than 15%).

# 4 Heuristics based on column generation

To obtain primal bounds, we adapt several classical heuristics to the context of a column generation approach. We provide a classification of such methodologies and a review of previous work where greedy, local search, rounding or other LP based heuristics have been used in a decomposition approach.

A natural way to obtain an integer solution is to solve the master restricted to the set of generated columns as an integer program. For this, we must use the discrete master formulation (note that each column $r$ generated for the aggregate master translates into a column $q$ for each feasible starting date $s$ in the discrete master). The rounding heuristic differs from the restricted master heuristic by the fact that new columns are generated in the course of the procedure but instead of fully exploring a branch-and-bound tree, a heuristic selection of a branch made at each node: a column of the LP solutions is rounded up and the master LP is re-optimized by column generation. For this implementation we use both the aggregate and the discrete master programs: LP solution are computed for the aggregate master; but the partial master solution is recorded in the discrete formulation (choosing a starting date for each column selected by the rounding procedure) and the columns used in the aggregate formulation are restricted to those that could be part of an integer solution to the residual discrete master program. Our third heuristic is a local search procedure where a neighbor solution is defined by removing a few columns from the current integer solution and re-building a complete integer solution with the rounding heuristic procedure.

Our computational experiments show that solving the restricted discrete master program to integer optimality is quite computational intensive (for instances with 150 customers we have no solution after 4 hours of computing time); this is partially due to symmetry. With the rounding heuristic, we obtain primal bounds whose optimality gap is around 10% for instances of industrial size (our test bed includes instances with up to 260 customers). By imposing some restrictions on solution space (such as further restricting the set $P$) we sometime get smaller optimality gaps. The local search procedure permits small improvements. We believe that much of our optimality gaps comes from the lack of strength of our dual bound.

# References

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

[2] AM. Campbell, LW Clarke, MWP. Savelsbergh, Inventory routing in practice, in The Vehicle Routing Problem, eds P. Toth, D. Vigo, SIAM Monographs on Discrete Mathematics and Applications, 2002.

[3] V. Gaur, ML. Fisher, A periodic inventory routing problem at a supermarket chain, *Operations Research*, 52(6):813–822,2004.

[4] D. Pisinger, A minimal Algorithm for the Multiple-Choice Knapsack Problem, *EJOR*, 83:394–410, 1995.

[5] F. Vanderbeck and M.W.P. Savelsbergh, A Generic View of Dantzig-Wolfe Decomposition in Mixed Integer Programming, *Operations Research Letters*, To appear.