Inventory Routing Problem Solved by Heuristic Based on Column Generation

Sophie Michel François Vanderbeck
University Bordeaux 1, 351 Cours de la Libération, 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. At the tactical level, the objective is to minimize the fleet size and an estimate of the distance travelled. Moreover, for practical purposes, routes must be geographically clustered and the planning must be repeated over the time horizon with constrained periodicity. We develop a truncated branch-and-price-and-cut algorithm combined with rounding and local search heuristics that yield both primal solutions and dual bounds. Periodic plannings are generated for vehicles by solving a multiple choice knapsack problem. The issues related to the construction of the customer plannings are dealt with in a master program. 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. Real-life instances of the problem are solved with reasonable optimality gaps.

Keywords: Real World Application, Vehicle Routing, Planning, Primal Heuristics.

Introduction

The Inventory Routing Problem (IRP) combines issue of vehicle 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, 5]. The existing approaches tend to make restrictive assumption (such as assuming a fixed partition policy) 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 consider real-life instances of the problem (using data coming from our industrial partner). In the application that motivates our study, the stock management policy is simple with deterministic consumption rate and an order-up-to-level policy. Section 1 describes this problem and our assumptions. The problem is well-suited for decomposition, hence our approach relies on Dantzig-Wolfe reformulation [10] as outlined in Section 2. The latter eliminates the symmetry in vehicle indexing (vehicles are identical) but still suffers from a symmetry in time (shifting the starting time of routes in a periodic solutions can define a symmetric solution). Hence, we model average behavior by considering a single aggregate variable measuring how many times a specific route and associated delivery pattern is used over all possible starting dates. Cutting planes are added to the master to improve the formulation. Dual bounds are obtained by LP relaxation and tightened through partial branching. Heuristics based on column generation give primal bounds. Sections 3 and 4 present methodology to obtain respectively dual and primal bounds as well as computational results. Primal heuristics developed here in a column generation context are generic and can be used for others problems.

1 The Problem

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 in our tests $P = \{1, 2, 3, 4, 5, 6\}$. 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 bounded by the least common multiple of the periodicities (in our example $H \leq T = 60$). T is the maximal 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 by some vehicle 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 and the specific quantities that are collected on each visited point (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 favor 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 that yields clusters of points that are geographically gathered around their seed.

2 A Dantzig-Wolfe Decomposition approach

The problem decomposes into planning issues on one hand and routing issues on the other. We formulate the planning problem in terms of variables associated with the selection of routes. The definition of a route specifies the visited customers, 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 the quantity that is collected is the production of ℓ periods; the associated profit is $p_{i\ell}$; the knapsack formulation is

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

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

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

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

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

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, that are defined as the solutions, ϕ^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 (ϕ^q, p^q, s^q), one can generate an indicator matrix δ^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:

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

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

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

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

$$Vmax \in \mathbb{N}. \tag{10}$$

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 λ_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 the columns: aggregating all columns that differ only by their starting dates, we project our column space as follows

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

At each column r is associated a set of columns q such that r is the projection of q:

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

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:

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

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

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

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

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

where λ_r is the 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.

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 consist in generating a periodic route: the knapsack problem (1-5) is solved for each periodicity and each seed. A dynamic program [7] returns the knapsack solution. To speed up the resolution time, a preprocessing is performed at each step (items with negative reduced cost are not considered and the dynamic program is called only if there is hope to find a negative reduced cost column). Moreover, the enumeration of the pricing subproblems associated with each pair (periodicity, seed) stops as soon as a column with negative reduced cost is found.

A cutting planes procedure is implemented based on a family of valid inequalities that we derived from (12) using a rounding procedure. Let h be an integer ranging from 1 to $T-1$ and $i \in N$ such that $tmax_i > 1$, the 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 only on variable $Vaver$. 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.

We have made comparative tests on real and randomly generated instances. We have 5 real instances with 60 customers on average, this group is named "S60", and 2 bigger instances with

172 and 157 customers, named GN172 and Pb157. We generate 10 random instances imitating the real problem (the customer coordinates are generated according to 3 schemes: urban, rural or mixed). This group is named "AL100" (we make these random instances available on our web site [11]). 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%). In the Table 1, we present the dual bound obtained at the root of the branch-and-price tree, "rootDB", and the bound obtained after branching and adding cut, "DB+br+cut". To compare these bounds, we compute the gap using a good primal solution (see section 4), and give the gap at root, "gap", after adding cut, "gap+cut", after branching, "gap+br", and after branching and adding cut, "gap+br+cut". The initial dual bound is improved by 12.42%.

Name	rootDB	gap	gap+cut	gap+br	DB+br+cut	gap+br+cut
av AL100	325.674	22.81	21.31	10.37	366.123	9.24
av S60	224.836	26.38	25.51	12.51	254.083	11.83
GN172	517.348	16.46	15.59	8.65	558.452	7.89
Pb157	601	15.40	14.39	5.83	659.839	5.11
av 17 inst		**23.05**	**21.80**	**10.63**		**9.67**

Table 1: Dual bounds on our test bed

4 Heuristics based on column generation

To obtain primal bounds, we adapt several classical heuristics to the context of a column generation approach. In [6], we provide a classification of such methodologies and a review of previous work (f.i., [3, 4, 8, 9]) 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 is made at each node: a column of the LP solution 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 computationally intensive (for instances with 100 customers we have no solution after 2 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. By imposing some restrictions on solution space (such as further restricting the set $P = \{1, 2, 3\}$) we sometime get smaller optimality gaps, a post-optimisation procedure gives the solution on the larger set $P = \{1, 2, 3, 6\}$. The local search procedure allows small improvements. In Table 2, we present the primal bound and the associated duality gap that we obtained by rounding heuristic called at root and after branching, "PB+RH" and "gap+RH", and by further applying the local search improvement procedure at the root node, "PB+LS" and "gap+LS", and the gap obtained in calling the rounding heuristics after restricting the set P to $\{1, 2, 3\}$ and making use of a post-optimisation heuristic, "gap+PO".

Name	PB+RH	gap+RH	PB+LS	gap+LS	gap+PO
av AL100	409.399	12.29	407.892	11.88	8.72
av S60	285.056	11.75	285.18	11.74	10.40
GN172	589.582	5.574	590.48	5.73	5.90
Pb157	731.48	10.85	708.594	7.39	5.11
av 17 inst		**11.66**		**11.22**	**8.84**

Table 2: Primal bounds on our test bed

5 Results for a large industrial instance

Finally, we applied our approach to a large real-life industrial instances with 260 customers. In 8h34m of computing time we obtained the dual bound, "DB+br+cut", of value 838.17 (5h21m was spent in the cutting plane procedure). If however one is only looking for a good primal solution, our primal heuristics can be run skipping the time consuming in cutting plane procedure. In a separate run (turning off the cutting plane procedure), we obtained in 2h53m a primal solution with gap of 6.23% (PB=890.384) by calling rounding heuristics followed by local search on restricted $P = \{1, 2, 3\}$ and by applying post-optimisation. If we compare this solution to the one that is currently used by our industrial partner, we observe that our solution requires 9 instead of 10 vehicles and reduces the travelling distance by more than 10 percent.

Conclusion

We consider a real-life application of the inventory routing problem on a tactical planning level. Our primal heuristics based on exact optimisation tools provides much better solutions than the greedy heuristic approaches that we originally tested. Our dual bounds allow us to provide a guarantee on the optimality gap. Moreover, comparison with the solution currently used by our industrial partner show significant improvements. The key to success is a formulation modelling an average behavior to avoid symmetry.

References

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

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

[3] A. Chabrier (2003), Heuristic Branch-and-Price-and-Cut to solve a network design problem, *Proceedings CPAIOR.*

[4] FL. Cimelire (2004), Optimisation du traitement de l'ordre de fabrication dans l'industrie textile, *Thesis, University Bordeaux 1.*

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

[6] S. **Michel** (2006). Optimisation des tournes de vhicules combine la gestion des stocks. Phd Thesis, University Bordeaux 1.

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

[8] E. Taillard (1999), A heuristic column generation method for the heterogeneous VRP, *RAIRO Operations Research* **33**, 1 – 14.

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

[10] F. Vanderbeck and M.W.P. Savelsbergh (2006), A Generic View of Dantzig-Wolfe Decomposition in Mixed Integer Programming, *Operations Research Letters* **34**(3), 296 – 306.

[11] IRP instances available at `http://www.math.u-bordeaux.fr/~smichel/IRPinstances.tar`