High-performance local search for solving real-life inventory routing problems

Thierry Benoist[1], Bertrand Estellon[2], Frédéric Gardi[1], Antoine Jeanjean[1]

[1] Bouygues e-lab, Paris, France

[2] Laboratoire d'Informatique Fondamentale – CNRS UMR 6166, Université Aix-Marseille II – Faculté des Sciences de Luminy, Marseille, France

{tbenoist,fgardi,ajeanjean}@bouygues.com, bertrand.estellon@lif.univ-mrs.fr

Abstract. In this paper, a real-life routing and scheduling problem encountered is addressed. The problem, which consists in optimizing the delivery of fluids by tank trucks on a long-term horizon, is a generalization of the vehicle routing problem with vendor managed inventory replenishment. The particularity of this problem is that the vendor monitors the customers' inventories, deciding when and how much each inventory should be replenished by routing tank trucks. Thus, the objective of the vendor is to minimize the logistic cost of the inventory replenishment for all customers over the long run. Then, an original local-search heuristic is presented for solving the short-term planning problem. The engineering of this algorithm follows the three-layers methodology for "high-performance local search" recently introduced by some of the authors. A computational study demonstrates that our solution is both effective, efficient and robust, providing long-term savings exceeding 20 % on average, compared to solutions computed by expert planners or even a classical greedy algorithm. The resulting software is now exploited in North America by one of the French industry leaders.

1 Presentation of the problem

The problem addressed in this paper is a real-life inventory routing problem (IRP) occurring in one of the world's leading companies in its field. For the sake of concision, the problem is not completely and formally described here, but its main characteristics are outlined.

Spread over a geographical area, some customers consume fluid products and plants produce it. Each customer is equipped with a storage; similarly, each plant has a storage from which product can be pumped. Reliable forecasts of production at plants are known over a short-term horizon. On the customer side, two kinds of resupply are managed by the vendor. The first one, called "forecasting-based resupply", corresponds to clients for which reliable forecasts of consumption are available over a short-term horizon. The inventory of each customer must be replenished by tank trucks so as to never fall under its safety level. The second one, called "order-based resupply", corresponds to customers which send orders to the vendor, specifying the desired quantity and the time window in which the delivery must be done. Some customers can ask for the both types of resupply management: their inventory is replenished by the vendor using monitoring and forecasting, but they keep the possibility of ordering (to deal with an unexpected increase of their consumption, for example). Constraints consisting in maintaining inventory levels above safety levels (no stock out) and in satisfying orders (no missed orders) are defined as soft, since the existence of an admissible solution is not ensured in real-life conditions.

The transportation is performed by vehicles composed of three kinds of heterogenous resources: drivers, tractors, trailers. Each resource is assigned to a base. A vehicle corresponds to the association of one driver, one tractor and one trailer. Some triplets of resources are not admissible (due to driving licences, for example). The availability of each resource is defined through a set of time windows. Each site (plant or customer) is accessible to a subset of resources (special skills or certifications are required to work on certain sites). Thus, scheduling a shift consists in defining: a base, a triplet of resources (driver, tractor, trailer), and a set of operations each one defined by a triplet (site, date, quantity) corresponding to the pickups or deliveries performed along the tour. A shift must start from the base to which are assigned the resources composing the vehicle and must end by returning to this base. The working and driving times of drivers are limited; as soon as a maximum duration is reached, the driver must take a rest with a minimum duration (Department of Transportation rules). In addition, the duration of a shift cannot exceed a maximal value depending on the driver. The sites visited along the tour must be accessible to the resources composing the vehicle. A resource can be used only during one of its availability time windows. The date of pickup/delivery must be contained in one of the opening time windows of the visited site. Finally, the inventory dynamics, which can be modeled by flow equations, must be respected at each time step, for each site inventory and each trailer. In particular, the sum of quantities delivered to a customer (resp. loaded at a plant) minus (resp. plus) the sum of quantities consumed by this customer (resp. produced by this plant) over a time step must be lower (resp. greater) than the capacity of its storage (resp. zero). Note that here the duration of an operation does not depend on the delivered or loaded quantity; this duration is fixed in function of the site where the operation is performed, the resulting approximation being covered by the uncertainties lying on the traveled times.

In our case, reliable forecasts (for both plants and customers) are known over a 15-days horizon. Thus, shifts are planned deterministically day after day with a rolling horizon of 15 days. It means that each day, a distribution plan is built for the next 15 days, but only shifts starting at the current day are fixed. The objective of the planning is to respect the soft constraints described above over the long run (satisfying orders, maintaining safety levels). In practice, the situations where these constraints cannot be met are extremely rare, because missed orders and stockouts are unacceptable for customers (on the other hand, safety levels must be finely tuned according to customer consumptions). Then, the second objective is to minimize over the long term a logistic ratio defined as the sum of the costs of shifts (which is composed of different terms related to the usage of resources) divided by the sum of the quantities delivered to customers. In other words, this logistic ratio corresponds to the cost per unit of delivered quantity.

Large-scale instances have to be tackled. A geographic area can contain up to 1500 customers, 50 sources, 50 bases, 100 drivers, 100 tractors, 100 trailers. All dates and durations are expressed in minutes (on the whole, the short-term planning horizon counts 21600 minutes); the inventory dynamics for plants and customers are computed with time steps of one hour (because forecasts are computed with this accuracy). The execution time for computing a short-term planning is limited to 5 minutes on standard computers.

2 Related works

Since the seminal work of Bell et al. [1] on a real-life inventory routing problem encountered at Air Products (a producer and distributor of industrial gases), a vast literature has emerged on the subject. In particular, a long series of papers was published by Savelsbergh et al. [2, 3, 5, 8, 9], motivated by a real-life problematic encountered at Praxair (another supplier of industrial gases). However, in many companies, inventory routing is still done by hand or supported by basic softwares, with rules like: serve "emergency" customers (that is, customers whose inventory is near to run out) using as many "full deliveries" as possible (that is, deliveries with quantity equal to the trailer capacity or, if not possible, to the customer tank capacity). For more references, the interested reader is referred to the recent papers by Savelsbergh and Song [8, 9], which give a good survey of the research done on the IRP over the past 25 years.

To our knowledge, the sole papers describing practical solutions for problems similar to the one addressed here are the ones by Savelsbergh et al. [3, 5, 8, 9]. The solution approaches described in these papers are the same in essence: the short-term planning problem is decomposed to be solved in two phases. In the first phase, it is decided which customers are visited in the next few days, and a target amount of product to be delivered to these customers is set. In the second phase, vehicle routes are determined taking into account vehicle capacities, customer delivery windows, drivers restrictions, etc. The first phase is solved heuristically by integer programming techniques, whereas the second phase is solved with specific insertion heuristics [4]. The experiments reported in the different works on the subject [1, 3, 5, 8, 9] show savings up to 10 % over the long run (with computation times of several minutes), compared to solutions obtained by a greedy algorithm based on the rules of thumb commonly used in practice (like the one cited above).

3 Contribution

To our acquaintance, no pure and direct local-search algorithm has been proposed for solving the IRP. A local-search approach is described by Lau et al. [7] for solving an inventory routing problem with time windows, but their approach is based on a decomposition scheme (distribution and then routing). In this paper, an original local-search heuristic is described for solving the short-term planning problem. We insist on the fact that no decomposition is done in our approach: the short-term planning is optimized directly over the 15-days horizon. This algorithm has been designed and engineered following the methodology described by Estellon et al. [6] in a companion paper. A computational study demonstrates that our solution is both effective, efficient and robust, providing long-term savings exceeding 20 % on average, compared to solutions computed by expert planners or even a classical greedy algorithm.

Following the methodology exposed in [6], our local-search heuristic is designed according to three layers. The first layer corresponds to the search strategy; here a first-improvement descent heuristic with stochastic selection of transformations is employed (an initial solution is computed using an urgency-based insertion heuristic). The second layer corresponds to the pool of transformations which defines the neighborhood; here more than one hundred transformations are defined on the whole, which can be grouped into a dozen of types (for operations: insertion, deletion, ejection, move, swap; for shifts: insertion, deletion, rolling, move, swap). Finally, the third layer, corresponding to the "engine" of the local search, consists of three main procedures common to all transformations: evaluate (which evaluates the gain provided by the transformation applied to the current solution), commit (which validates the transformation by updating the current solution and the associated data structures), rollback (which clears all the data structures used to evaluate the transformation). Since the duration of an operation does not depend on the quantity loaded or delivered, the evaluation procedure is separated into two routines: first the scheduling of shifts and then the assignment of volumes. These routines, whose running time is critical for performance, relies on incremental algorithms supported by special data structures for exploiting invariants of transformations.

The whole algorithm was implemented in C# 2.0 programming language (for running on Microsoft .NET 2.0 framework). The resulting program includes nearly 30000 lines of code, whose 6000 lines (20 %) are dedicated to check the validity of all incremental data structures at each iteration (only active in debug mode). The whole project (specifications, research, implementation, tests), realized during the year 2008, required nearly 300 man-days. All statistics and results presented here have been obtained on a computer equipped with a Windows Vista operating system and a chipset Intel Xeon X5365 64 bits (CPU 3 GHz, L1 cache 64 Kio, L2 cache 4 Mio, RAM 8 Go). The local-search algorithm attempts more than 10000 transformations per second, even for large-scale instances (thousand sites and hundred resources). Then, our algorithm visits *nearly 10 million solutions in the search space during 5 minutes of running time* (which is the desired time limit in operational conditions). When planning over a

15-days horizon, the memory allocated by the program does not exceed 30 Mo for medium-size instances (hundred sites, ten resources), and 300 Mo for large-scale instances (thousand sites, hundred resources). Note that the running time of the urgency-based insertion heuristic is of few seconds for large-scale instances. The acceptance rate, which corresponds to the number of accepted transformations (that is, not strictly improving current solution) over the number of attempted ones, varies essentially between 1 and 10 % according to instances and optimization phases. Note that this rate is quasi constant all along the search (that is, during the 5 minutes of running time), allowing a large diversification of the search without the use of metaheuristics. On the other hand, the number of (strictly) improving transformations is of several hundreds.

The local-search algorithm has been extensively tested on several dozens of benchmarks with different characteristics: realistic (that is, matching the operational conditions), pathological (for example, with plants whose production is stopped several days), large-scale (for example, with 1500 sites and 300 resources). On 5 long-term real-life benchmarks (105 days), the gain obtained by the local-search algorithm with 5 minutes of running time per planning iteration reaches 21.8 % (resp. 25.3 %) on average compared to solutions obtained by the urgency-based insertion heuristic (resp. solutions built by the logistic experts of the company for which this R&D project was conducted).

References

1. W. Bell, L. Dalberto, M. Fisher, A. Greenfield, R. Jaikumar, P. Kedia, R. Mack, P. Prutzman (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces* 13(6), pp. 4–23.
2. A. Campbell, L. Clarke, A. Kleywegt, M. Savelsbergh (1998). The inventory routing problem. In *Fleet Management and Logistics* (T. Crainic, G. Laporte, eds), pp. 95–113. Kluwer Academic Publishers, Norwell, MA.
3. A. Campbell, L. Clarke, M. Savelsbergh (2002). Inventory routing in practice. In *The Vehicle Routing Problem* (P. Toth, D. Viego, eds), pp. 309–330. SIAM Monographs on Discrete Mathematics and Applications 9. SIAM, Philadelphia, PA.
4. A. Campbell, M. Savelsbergh (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science* 38(3), pp. 369–378.
5. A. Campbell, M. Savelsbergh (2004). A decomposition approach for the inventory-routing problem. *Transportation Science* 38(4), pp. 488–502.
6. B. Estellon, F. Gardi, K. Nouioua (2009). High-performance local search for task scheduling with human resource allocation. (to appear *SLS 2009, the 2nd International Workshop on Engineering Stochastic Local Search Algorithms*)
7. H. Lau, Q. Liu, H. Ono (2002). Integrating local search and network flow to solve the inventory routing problem. In *Proceedings of AAAI 2002, the 18th National Conference on Artificial Intelligence*, pp. 9–14. AAAI Press, Menlo Park, CA.
8. M. Savelsbergh, J.-H. Song (2007). Inventory routing with continuous moves. *Computers and Operations Research* 34(6), pp. 1744–1763.
9. M. Savelsbergh, J.-H. Song (2008). An optimization algorithm for the inventory routing with continuous moves. *Computers and Operations Research* 35(7), pp. 2266–2282.