A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem

Claudia Archetti* Alain Hertz† Maria Grazia Speranza*

*Dipartimento di Metodi Quantitativi, Universitá degli Studi di Brescia
50, C. da S. Chiara,
25122 Brescia, Italia
{archetti,speranza}@eco.unibs.it

†Département de Mathématiques et de Génie Industriel, Ecole Polytechnique de Montréal
C.P. 6079, Succursale Centre-Ville
Montreal, H3C 3A7 Canada
alain.hertz@gerad.ca

1 Introduction

We consider the *Split Delivery Vehicle Routing Problem* (SDVRP) where a fleet of homogeneous vehicles has to serve a set of customers. Each customer can be visited more than once, contrary to what is usually assumed in the classical *Vehicle Routing Problem* (VRP) and the demand of each customer can be greater than the capacity of the vehicles. No constraint on the number of available vehicles is considered. There is a single depot for the vehicles and each vehicle has to start and end its tour at the depot. The objective is to find a set of vehicle routes that serve all the customers such that the sum of the quantities delivered in each tour does not exceed the capacity of the vehicles and the total distance travelled is minimized.

The SDVRP is a variant of the *Capacitated Vehicle Routing Problem* (CVRP) which is well known in the literature (for a survey of vehicle routing problems, see [15]). In [11] the authors have described a tabu search algorithm for the capacitated vehicle routing problem showing that this heuristic works well on this problem.

The SDVRP has been introduced in the literature only a few years ago. In [7] and [8] Dror and Trudeau have analyzed the savings generated by allowing split deliveries in a vehicle routing problem and they have presented a heuristic algorithm for the problem. They have shown that when the distances satisfy the triangle inequality there exists an optimal solution for the SDVRP where no pair of tours has two or more vertices in common. Valid inequalities for the SDVRP are described in [6] while real applications of the problem are studied in [13] and [14]. In [4] a lower bound is proposed for the SDVRP where the demand of each customer is lower than the capacity of the vehicles and the quantity delivered by the vehicles when visiting a customer is an integer number. In [9] the authors present a mathematical formulation and a heuristic algorithm for the SDVRP with grid network distances and time windows constraints. In [1] and [2] the authors have analyzed the k-SDVRP where each vehicle has a capacity equal to a given integer k, and where the demands of the customers as well as the quantity delivered by a vehicle when visiting a customer are integer numbers. They have proved that the problem is NP-hard when $k \geq 3$ and they have shown that, under specific conditions on the distances, the problem is reducible in polynomial time to a new problem where each customer has a demand that is lower than the capacity of the vehicles, with a possible reduction on the number of customers.

A *direct trip* in a k-SDVRP is a tour where a vehicle starts from the depot, goes directly to a customer where it delivers k units, and then turns back directly to the depot. Given an instance I of the k-SDVRP, one can build a *reduced instance*, denoted I_R, by modifying the demand d_i of each customer to $d_i - k \left\lfloor \frac{d_i}{k} \right\rfloor$. A solution s_R for I_R can then be transformed into a solution s for I by adding $\left\lfloor \frac{d_i}{k} \right\rfloor$ direct trips for each customer i. Now, given an instance I of the k-SDVRP, consider the algorithm that first determines an optimal solution s_R^* for the reduced instance I_R, and then builds the associated solution s^* for I. It is proved in [2] that this algorithm gives a worst case error of $\frac{3}{2}$ when the distances satisfy the triangle inequality.

In the next section, we describe a tabu search algorithm for the k-SDVRP. Some computational results are given in Section 3.

2 A tabu search algorithm for the k-SDVRP

It is not difficult to build instances for which Dror and Trudeau's algorithm [7]. cannot find the optimal solution. In this section we present a tabu search algorithm for the k-SDVRP, called SPLITABU, that avoids such a situation (i.e., all optimal solutions can be reached by our tabu search). SPLITABU is a very simple algorithm, easy to implement, where there are only two parameters to be set: the length of the tabu list and the maximum number of iterations the algorithm can run without improvement of the best solution found. The algorithm is composed of the three following phases.

- *Phase 1:* construction of an initial feasible solution.
- *Phase 2:* tabu search phase.
- *Phase 3:* final improvement of the solution found by the tabu search phase.

2.1 Phase 1 : construction of an initial feasible solution

For constructing an initial feasible solution, we first create a reduced instance by creating as many direct trips as possible (see Section 1). We then solve a traveling salesman problem on the reduced instance and we cut this giant tour into pieces so that the capacity constraints are satisfied. For building a giant TSP tour, we use the GENIUS algorithm proposed by Gendreau, Hertz and Laporte [10]. GENIUS is composed of two procedures: the first one, GENI, is a generalized insertion procedure and the second one, US, is a postoptimization routine. It is shown in [10] that this algorithm is a very efficient solution method for the traveling salesman problem.

2.2 Phase 2 : tabu search phase

The tabu search phase is a standard tabu search algorithm that stops when n_{max} iterations have been performed without improvement of the best solution encountered so far. According to preliminary experiments we have fixed $n_{\max}$ equal to $400n$, where n is the number of customers.

A move from a solution s to a neighbour solution s' is performed by including a customer i into a route r and by removing i from a set of routes visiting i. The insertion of a customer i in a route r is made with the classical cheapest insertion method. When a customer i is added to a route r, we consider as tabu for θ iterations the removal of i from r, and we will say that route r is tabu for i. Also, when a customer i is removed from a route u, then it is tabu for θ iterations to reinsert i into u, and we will also say that u is tabu for i. We have observed that values of θ that depend on the number n of customers and on the number g of routes in the current solution s produce better solutions. According to preliminary experiments, we have decided to set θ equal a random integer number in the interval $[\sqrt{10}, \sqrt{10+p}]$ where $p = n+g$ if $n+g < 100$ and $p = \frac{3}{2}(n+g)$ if $n+g \geq 100$, since this interval generates values of θ which are high enough to prevent cycling and not to high to prevent the algorithm from obtaining a good solutions.

The tabu restrictions defined above may be too strong and forbid a good neighbour solution. For this reason, we also consider the possibility of removing a customer i from routes that are tabu for i and to insert i into a route r which is tabu for i. However, such a neighbour is only accepted if it leads to a better solution than the best one encountered so far.

When all routes visiting a customer i are tabu, then we also consider the possibility of removing part of the demand of i from a route u, and inserting i into a route r. This cannot lead to a solution of better value than s and this is the reason why we do not allow such a move if r is tabu for i.

2.3 Phase 3 : final improvement

Dror and Trudeau have shown that, if the distances satisfy the triangle inequality, then there always exists an optimal solution to the SDVRP which does not contain t-split cycles with $t \geq 2$, where a t-split cycle is a set of routes $r_1, ..., r_t$ such that r_w contains customers i_w and i_{w+1}, $w = 1, ..., t-1$, and r_t contains customers i_1 and i_t. A t-split cycle can easily be removed from a solution as follows. Let q_w denote the quantity delivered to i_w on r_w, and let w^* be an index such that $q_{w^*} \leq q_w$ $w = 1, ..., t$: one can transfer q_{w^*} units of demand of each customer i_w, $w = 1, ..., t-1$, from r_w to r_{w+1} as well as the same quantity for customer i_t from r_t to r_1. Customer i_{w^*} will thus be removed from route r_{w^*}. If the distances satisfy the triangle inequality, then this new solution is possibly better than the one with the t-split cycle.

The final improvement phase performs such kind of improvements. It also tries to reduce the length of each route by applying the GENIUS algorithm.

3 Computational results

SPLITABU was implemented in C++ on a PC Pentium 4, 256 MB RAM. We have first generated problems of small size as follows. We have considered the n first customers of the benchmark VRP problems described in [11], with $n = 6, 7, \ldots, 15$. All these problems have a fixed vehicle capacity which was reduced to a smaller value so that at least three vehicles are needed in a VRP solution. These small problems were solved using CPLEX 6.6. While problems with up to 10 customers could be solved in a few seconds, larger problems with $n = 11, \ldots, 15$ required between one hour and four days of computation. For comparison, SPLITABU has produced the same optimal solutions for all these problems in less than one second.

Larger problems, with more than 15 customers can hardly be solved to optimality. In order to evaluate the performance of SPLITABU on such larger instances we compare the results produced by SPITABU with those obtained using Dror and Trudeau's algorithm (DT for short). Algorithm DT was implemented in C++ on the same PC. We have considered problems 1-5, 11 and 12 from [11]. These problems have between 50 and 199 customers. As proposed by Dror and Trudeau [7], the demands of the customers have been modified as follows. Let α and γ be two parameters chosen in the interval $[0, 1]$, with $\alpha \leq \gamma$. The demand d_i of customer i is chosen randomly in the interval $[\alpha k, \gamma k]$. As in [7], we have considered the following combinations (α, γ) of parameters : $(0.01, 0.1)$, $(0.1, 0.3)$, $(0.1, 0.5)$, $(0.1, 0.9)$, $(0.3, 0.7)$ and $(0.7, 0.9)$. We have also considered the case where the original demands are not changed. This gives a total of 49 instances (since there are 7 instances with different demands for each of the 7 VRP problems taken from [11]).

The algorithm proposed by Dror and Trudeau [7] uses the two following improvement procedures : procedure NODE INTERCHANGES [5] that performs one-node moves and two-nodes swaps between routes, and the classical 2-OPT procedure [12]. Preliminary experiments with SPLITABU have shown that some solutions can easily be improved by applying these two procedures each time the best solution s^* encountered so far is improved. This variant of SPLITABU is called SPLITABU-DT. Finally, since algorithm DT is much faster than our tabu search, we have considered a variant of SPLITABU-DT, called FAST-SPLITABU where Phase 2 is run for at most one minute.

Each variant of SPLITABU was run 5 times on each instance (two executions on a same instance may differ due to the randomness of the length of the tabu list). Detailed tables of results appear in [3]. We summarize here below our main observations. We have first noticed that DT is very fast since it requires less than one second for 35 of the 49 instances. In average, SPLITABU and SPLITABU-DT require less than 10 minutes for 35 and 31 instances, respectively. They both require more than one hour in only two cases. CPU times increase not only with the number of customers but also with their demands. Indeed, an instance with (α, γ)=$(0.01, 0.1)$ is typically solved much faster than the same instance with (α, γ)=$(0.7, 0.9)$. The reason is that more vehicles are needed when the demands are becoming larger, and this induces an increase in the number of neighbour solutions to be considered at each step of the tabu search.

Our next observation is that SPLITABU-DT finds better solutions than DT on all 49 instances. The improvement even reaches 17.34% on some instances. Algorithms SPLITABU and FAST-

SPLITABU both outperform DT on 43 instances. The advantage of the three variants of SPLITABU over algorithm DT is particularly visible on instances with small demands. More detailed results and comments can be found in [3].

4 Conclusions

SPLITABU is a very simple algorithm for the k-SDVRP, easy to implement, with only two parameters to be set. Computational experiments confirm that optimal solutions can be obtained in a practically null time for small instances having up to 15 customers. Comparison with Dror and Trudeau's algorithm (the only existing heuristic algorithm for the k-SDVRP) show that the variants of SPLITABU provide almost always better solutions even when computational times are limited to one minute. Another important issue is that one can easily build instances for which Dror and Trudeau's algorithm cannot find the optimal solution, while the neighbourhood defined in SPLITABU overcomes this difficulty.

References

[1] C. Archetti, R. Mansini, M.G. Speranza, "The split delivery vehicle routing problem with small capacity", Technical report n. 201, Department of Quantitative Methods, University of Brescia, 2001 (submitted).

[2] C. Archetti, M.G. Speranza, "A direct trip algorithm for the k-split delivery vehicle routing problem", Technical report n. 205, Department of Quantitative Methods, University of Brescia, 2002 (submitted).

[3] C. Archetti, A. Hertz, M.G. Speranza, "A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem", to appear in *Transportation Science,* 2003.

[4] J.M. Belenguer, M.C. Martinez, E. Mota, "A lower bound for the split delivery vehicle routing problem", *Ops. Res.,* vol. 48, 801-810, 2000.

[5] M. Dror, L. Levy, "A vehicle routing improvement algorithm comparison of a "greedy" and a matching implementation for inventory routing", *Computers Ops. Res.*, vol.13, 33-45, 1986.

[6] M. Dror, G. Laporte, P. Trudeau, "Vehicle routing with split deliveries", *Discrete Appl. Math.,* vol. 50, 239-254, 1994.

[7] M. Dror, P. Trudeau, "Savings by split delivery routing", *Transp. Science,* vol. 23, 141-145, 1989.

[8] M. Dror, P. Trudeau, "Split delivery routing", *Naval Res. Logist.,* vol. 37, 383-402, 1990.

[9] P.W. Frizzell, J.W. Giffin, "The split delivery vehicle scheduling problem with time windows and grid network distances", *Computers Ops. Res.,* vol. 22, 655-667, 1995.

[10] M. Gendreau, A. Hertz, G. Laporte, "New insertion and postoptimization procedures for the traveling salesman problem", *Ops. Res.,* vol. 40, 1086-1094, 1992.

[11] M. Gendreau, A. Hertz, G. Laporte, "A tabu search heuristic for the vehicle routing problem", *Manag. Science,* vol. 40, 1276-1290, 1994.

[12] S. Lin, "Computer solutions of the traveling salesman problem" *Bell System Tech. J.*, vol. 44, 2245-2269, 1965.

[13] P.A. Mullaseril, M. Dror, J. Leung, "Split-delivery routing in livestock feed distribution", *J. of Operational Res. Society,* vol. 48, 107-116, 1997.

[14] G. Sierksma, G.A. Tijssen, "Routing helicopters for crew exchanges on off-shore locations", *Ann. Oper. Res.,* vol. 76, 261-286, 1998.

[15] P. Toth, D. Vigo (eds.), "*The Vehicle Routing Problem*", SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002).