Les Cahiers du GERAD ISSN: 0711–2440

One-to-Many-to-One Single Vehicle Pickup and Delivery Problems

I. Gribkovskaia
G. Laporte

G–2007–04

January 2007

One-to-Many-to-One Single Vehicle Pickup and Delivery Problems

Irina Gribkovskaia

Molde University College
Postbox 2110
N-6402 Molde, Norway
irina.gribkovskaia@himolde.no

Gilbert Laporte

GERAD and
Canada Research Chair in Distribution Management
HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada, H3T 2A7
gilbert@crt.umontreal.ca

January 2007

Les Cahiers du GERAD

G–2007–04

Abstract

In One-to-Many-to-One Single Vehicle Pickup and Delivery Problems a vehicle based at the depot must make deliveries and pickups at customers locations before returning to the depot. Several variants can be defined according to the demand structures and sequencing rules imposed on pickups and deliveries. In recent years there has been an increased interest in this family of problems. New formulations and efficient heuristics capable of yielding general solutions (unrestricted in shape) have been proposed. In addition, some new and interesting extensions have been analyzed, including problems with selective pickups and problems with capacitated customers. The purpose of this chapter is to review these developments.

Key Words: Pickups and deliveries, clustered traveling salesman problem, backhauls, lasso, double-path, general solutions, reverse logistics, selective pickups, transshipment, capacitated customers.

Résumé

Dans les problèmes de cueillette et livraison de type "un-à-plusieurs-à-un", un véhicule basé au dépôt doit effectuer des cueillettes et livraisons chez un ensemble de clients avant de retourner au dépôt. On peut définir plusieurs variantes de ces problèmes selon la structure de la demande et les règles de séquencement imposées sur les cueillettes et les livraisons. Au cours des dernières années, on a assisté à un intérêt accru pour ces problèmes. En particulier, on a proposé de nouvelles formulations donnant lieu à des solutions dont la forme n'est pas contrainte (des solutions dites générales). De plus, plusieurs nouvelles extensions intéressantes ont été analysées, incluant des problèmes avec cueillettes sélectives ou avec clients de capacité limitée. Le but de ce chapitre est de passer ces développements en revue.

Mots clés : cueillettes et livraisons, problème du voyageur de commerce avec groupes de clients, cueillettes de retour, lasso, chemins doubles, solutions générales, logistique inverse, cueillettes sélectives, transbordement, clients à capacité limitée.

Acknowledgments: This work was partly funded by the Research Council of Norway and by the Natural Sciences and Engineering Research Council of Canada under grant 39682-05. This support is gratefully acknowledged.

1 Introduction

One-to-Many-to-One Single Vehicle Pickup and Delivery Problems (1-M-1 SVPDPs) are defined on a graph $G = (V, A)$, where $V = \{0, 1, \ldots, n\}$ is a vertex set and $A = \{(i, j) : i, j \in V, i \neq j\}$ is an arc set. Vertex 0 is a depot while the remaining vertices are customers. A vehicle of capacity Q is based at the depot. Each customer i has a pickup demand p_i and a delivery demand d_i satisfying $p_i \geq 0$, $d_i \geq 0$, $\sum_{i=1}^{n} p_i \leq Q$ and $\sum_{i=1}^{n} d_i \leq Q$. A non-negative cost c_{ij} is associated with each arc (i, j). The aim is to construct a least cost route starting and ending at the depot, and making all pickups and deliveries without ever exceeding the vehicle capacity. We assume that pickup and delivery demands are unsplittable and that no transshipments are allowed.

In the 1-M-1 SVPDP, the expression "one-to-many-to-one" means that all delivery demands are initially located at the depot, and all pickup demands are destined to the depot. Taken collectively, all delivery demands can be viewed as a single commodity, and all pickup demands can be viewed as a second commodity. These problems are different from many-to-many (M-M) problems, like the *Swapping Problem* (Anily and Hassin, 1992) in which commodities of several types have to be shifted among vertices, and from one-to-one (1-1) problems, like the *Stacker Crane Problem* (Frederickson, Hecht and Kim, 1978) in which commodities must be moved between specific origin-destination pairs.

Applications of 1-M-1 SVPDPs arise in several reverse logistics operations involving, for example, in the delivery of full bottles and the collection of empty ones (Dethloff, 2001; Tang and Galvão, 2002, 2006; Privé et al., 2006), in mail services (Wasner and Zäphel, 2004), and in the servicing of offshore platforms (Gribkovskaia, Laporte and Shlopak, 2006).

It is convenient to distinguish between two variants of 1-M-1 SVPDPs. In the first variant, denoted by P/D, and referred to as the *SVPDP with single demands*, each customer i has a positive pickup or a positive delivery demand, but not both, i.e., $p_i = 0$ or $d_i = 0$. In the second variant, denoted by P&D, customers may have positive pickup and delivery demands. We will refer to this variant as the *SVPDP with combined demands*.

In recent years, several new algorithms and applications have been proposed for 1-M-1 SVPDPs. A number of interesting properties have also been identified. While the problems can readily be formulated as mixed integer linear programs, only relatively small instances can be solved optimally with such formulations. Most research has therefore been devoted to the development of heuristics. These include construction and improvement schemes based on classical mechanisms, and more powerful methods based on metaheuristics, almost exclusively tabu search. In a number of cases, heuristics with a bounded worst-case performance ratio have been put forward.

Our purpose is to review these developments with an emphasis on theoretical properties and tabu search. The remainder of this chapter is organized as follows. Sections 2 and 3 are devoted to the SVPDP with single demands and to the SVPDP with combined demands, respectively. Some extensions of the models of Section 3 are presented in Section 4, followed by conclusions in Section 5.

2 The SVPDP with single demands (P/D)

In the SVPDP with single demands, two cases are possible. In the *SVPDP with backhauls*, all delivery customers must be visited before pickup customers. In the *mixed SVPDP*, no a priori sequence is imposed.

2.1 The SVPDP with backhauls

The SVPDP with backhauls is more commonly known as the *Traveling Salesman Problem with Backhauls* (TSPB). In this problem, customers with delivery demands are called *linehaul* customers, while customers with pickup demands are called *backhaul* customers. The TSPB is essentially a *Clustered TSP* (CTSP) (Chisman, 1975) with the three clusters: $\{0\}, D = \{i \in V : d_i > 0 \text{ and } p_i = 0\}$, and $P = \{i \in V : p_i > 0 \text{ and } d_i = 0\}$. As suggested by Chisman (1975), the CTSP can be transformed into a *Traveling Salesman Problem* (TSP) by adding an arbitrarily large constant M to the cost of all arcs linking any two of the sets $\{0\}$, D and P.

We are interested in the case where the costs c_{ij} are symmetric, so that each pair of arcs $\{(i,j),(j,i)\}$ can be replaced with a single edge (i,j), where $i < j$. In the case of symmetric costs satisfying the triangle inequality, Chisman's transformation preserves this property and allows the application of the Christofides (1976) heuristic to the transformed instance. While this heuristic has a worst-case ratio of 1.5 for the TSP, it yields a useless bound for the TSPB (Gendreau, Hertz and Laporte, 1997). However, a heuristic with a worst-case performance ratio of 1.5 for the TSPB can still be constructed as follows:

Step 1. Construct a spanning tree S of G whose edge set consists of (1) the edges of S_D, a minimum cost spanning subtree of the graph induced by D, (2) edge $(0,d)$ with $c_{0d} = \min_{j \in D}\{c_{0j}\}$, (3) edge $(0,p)$ with $c_{0p} = \min_{j \in P}\{c_{0j}\}$, (4) the edges of S_P, a minimum cost spanning tree of the graph induced by P.

Step 2. Let R be the set of odd-degree vertices in S. (Note that $0 \notin R$, $|R|$ is even, $|R \cap D|$ is odd, and $|R \cap P|$ is odd). Determine a minimum cost matching H on the edges of the graph induced by R with respect to the transformed costs.

Step 3. Construct a Eulerian subgraph using the edges of $S \cup H$. This graph contains edges $(0,d)$, $(0,p)$ and a single edge $(\bar{d},\bar{p})$ between D and P. Using a shortcut technique (Christofides, 1976), extract from this Eulerian graph a Hamiltonian chain $(d,\ldots,\bar{d})$ on the graph induced by D, linking a Hamiltonian chain $(\bar{p},\ldots,p)$ on the graph induced by P. The tour $T = (0,d,\ldots,\bar{d},\bar{p},\ldots,p,0)$ is a feasible TSPB solution.

Gendreau, Hertz and Laporte (1997) show that if z is the cost of T and z^* is the optimal TSPB cost, then $z/z^* \leq 1.5$ and this bound is tight. The authors have applied this heuristic to randomly generated instances and have shown that it yields average deviations of 30% but this value can be brought down to 5% by applying a postoptimization phase. Mladenović and Hansen (1997) have later improved this value slightly by means of a variable neighbourhood search heuristic.

2.2 The mixed SVPDP

The mixed SVPDP has been called the *TSP with Pickups and Deliveries* by Mosheiov (1994), the *TSP with Delivery and Backhauls* by Anily and Mosheiov (1994), and the *Mixed TSP* by Nagy and Salhi (2005).

An interesting result due to Mosheiov (1994) is that a feasible solution of the mixed SVPDP always exists provided $\sum_{i=1}^{n} p_i \leq Q$ and $\sum_{i=1}^{n} d_i \leq Q$. Such a solution can be generated as follows:

Step 1. Construct a Hamiltonian circuit $(i_1, \ldots, i_n, i_1)$ on the graph induced by $V \backslash \{0\}$, disregarding pickup and delivery demands.

Step 2. Let i_r be such that $\sum_{t=1}^{r} (p_{i_t} - d_{i_t})$ is maximized.

Step 3. Then, the Hamiltonian circuit $(0, i_{r+1}, \ldots, i_n, i_1, \ldots, i_r, 0)$ is feasible.

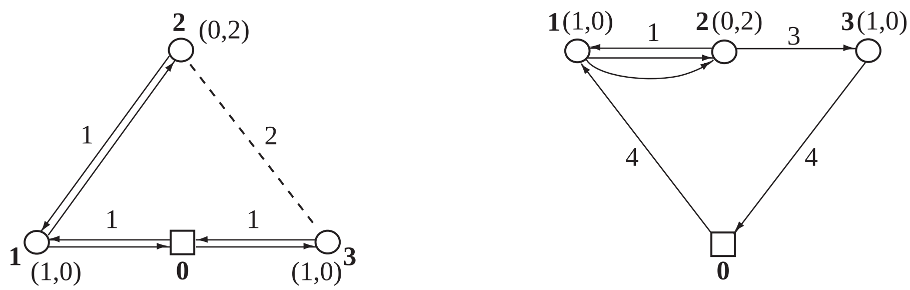
Moreover, if a TSP algorithm with a worst-case performance ratio of α is used in Step 1, then the worst-case performance ratio of the algorithm just described is $1+\alpha$. The algorithm is called PDαT. If the Christofides (1976) algorithm is used in Step 1 (assuming costs are symmetric and satisfy the triangle inequality), then the overall algorithm has a worst-case performance ratio of 2.5.

Mosheiov (1994) presented another heuristic, based on that of Golden et al. (1980) with an unbounded worst-case performance ratio but a better empirical performance. It first constructs a TSP solution on D and then gradually inserts the vertices of P using a cheapest insertion criterion, while maintaining feasibility.

Anily and Mosheiov (1994) have later proposed another heuristic called 2MST. In what follows, the *net demand* of a subtree is the total demand of its pickup vertices, minus the total demand of its delivery vertices.

Step 1. Compute a minimum spanning tree on G.

Step 2. Starting at vertex 0, traverse the tree in a depth-first fashion, visiting first the subtrees with a positive net demand. Vertices with a delivery demand are served the first time they are visited, while vertices with a pickup demand are served after all vertices in the subtree rooted at them have been served.



a) All edges but one are traversed twice ($Q = 2$)

b) All edges are traversed once or three times ($Q = 2$)

Figure 1: Two possible solutions (full lines) to the mixed SVPDP on a circular graph. The pickup and delivery amounts are indicated by (p_i, d_i).

Step 3. Construct a Hamiltonian tour by following the tree in a depth-first fashion and applying shortcuts.

The authors have proved that the solution produced by this heuristic is always feasible. Moreover, if the cost matrix is symmetric and satisfies the triangle inequality, then the worst-case performance ratio of heuristic 2MST is equal to 2.

Another heuristic for the same problem was proposed by Gendreau, Laporte and Vigo (1999). The authors first consider the SVPDP defined on a cycle. They show that there always exists an optimal solution in which one edge is not visited and all other edges are visited twice, or all edges are visited once or three times. In the first case the edge having the highest cost is unvisited. In the second case, assuming the order of the vertices on the cycle is $(0, 1, \ldots, n)$, then edge $(i, i+1)$ is visited three times if and only if $\sum_{j \leq i}(p_j - d_j) > Q - \sum_{j=1}^{n} d_j$. The authors provide a linear time exact algorithm called C, for visiting each edge the correct number of times. Figure 1 depicts two solutions for the mixed SVPDP.

Given this, the following $O(n^2)$ heuristic, called H, can be applied to a general graph.

Step 1. Determine a TSP solution on G by means of the Christofides (1976) heuristic.

Step 2. Apply heuristic C to the Hamiltonian cycle corresponding to the TSP solution.

Step 3. Derive a Hamiltonian solution by applying shortcuts.

Since the cost of this solution is at most twice that of the TSP solution and the Christofides heuristic has a worst-case performance ratio of 1.5, heuristic H has a worst-case performance ratio of 3. While this ratio is larger than that of the Mosheiov (1994) heuristic, it has been shown to have a better empirical performance. Gendreau, Laporte and Vigo (1999) have also developed a tabu search heuristic that performs 2-opt exchanges, which outperformed all previous heuristics at the expense of larger computing times.

Finally, we mention that Baldacci, Hadjiconstantinou and Mingozzi (2003) have developed an exact branch-and-cut algorithm for this problem, based on a two-commodity flow formulation. It can solve instances involving up to 200 vertices within one hour of computing time.

3 The SVPDP with combined demands (P&D)

We distinguish between two cases of the SVPDP with combined demands. In the first case, denoted by PD and known as the *Traveling Salesman Problem with Pickup and Deliveries* (TSPPD), each customer is visited exactly once for a combined pickup and delivery operation. It will be shown that this case reduces to the mixed SVPDP. In the second case, denoted by P-D, and called the *general SVPDP*, the pickup and delivery operations may be performed within the same visit or in two separate visits.

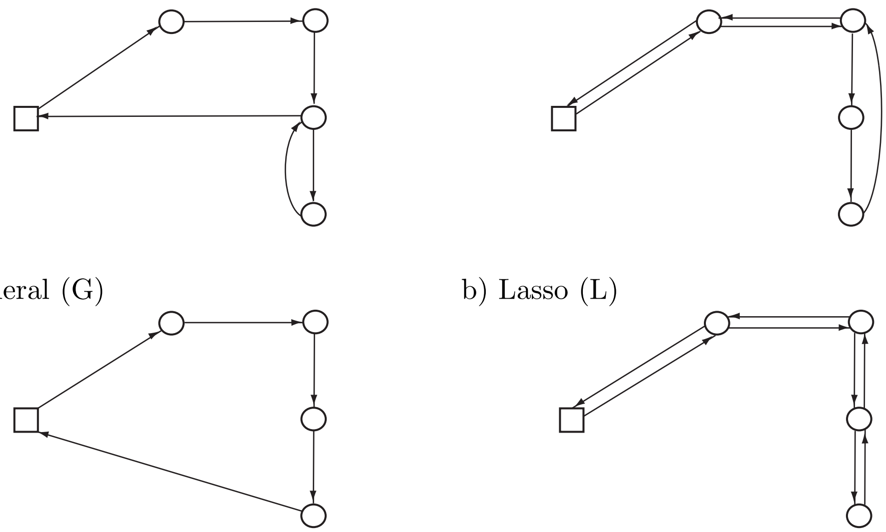
3.1 The Traveling Salesman Problem with Pickup and Deliveries (PD)

The TSPPD reduces to the mixed SVPDP. Indeed, if $p_i > d_i$, redefine the pickup demand of customer i as $p'_i = p_i - d_i$, and its delivery demand as $d'_i = 0$; if $p_i \leq d_i$, the delivery demand of i becomes $d'_i = d_i - p_i$ and its pickup demand is $p'_i = 0$. Then redefine the vehicle capacity as $Q' = \max\left\{\sum_{i=1}^{n} p'_i, \sum_{i=1}^{n} d'_i\right\}$. Note that this transformation is only valid under the assumption that each customer is visited only once, which makes it possible to work with net demands. All methods of Section 2.2 are applicable to this problem.

3.2 The general SVPDP (P-D)

Gribkovskaia et al. (2006) distinguish between four solution shapes for the general SVPDP: general (G), lasso (L), Hamiltonian (H), and double-path (D). A general solution is unrestricted in that any customer can be visited once for a combined pickup and delivery service, or twice if these two operations are performed separately. In a lasso solution, the vehicle first performs deliveries along a path rooted at the depot to a subset S of customers, until it reaches a certain vertex k. All vertices of $(V\backslash\{0\})\backslash S$ are then visited once for a combined service along a loop until the vehicle reaches k again and performs deliveries to the customers of S by following a path leading to the depot. If $S = \emptyset$, the lasso reduces to a Hamiltonian solution, which yields a TSPDP. If $S = V\backslash\{0\}$, the lasso reduces to a double-path solution. A double-path solution can also be obtained by solving a TSPB. This is achieved by duplicating the customer set into the union of a set of linehaul customers with delivery demands d_i and zero pickup demands, and a set of backhaul customers with zero delivery demands and pickup demands p_i. The four solution shapes are illustrated in Figure 2.

Denote by z_G, z_L, z_H and z_D the costs of the optimal general, lasso, Hamiltonian and double-path solutions, respectively, associated with the same instance. Gribkovskaia et al.



a) General (G) b) Lasso (L)

c) Hamiltonian (H) d) Double path (D)

Figure 2: Four solution shapes for the general SVPDP with combined demands on a Euclidean graph.

(2006) prove that if the (c_{ij}) matrix satisfies the triangle inequality, then the following relation holds: $z_G \leq z_L \leq z_H \leq z_D \leq 2z_G$. Figure 3 depicts an instance for which the non-lasso solution $(0, 1, 2, 3, 5, 4, 6, 3, 0)$ of cost 9 is optimal.

The general SVPDP with combined demands can be formulated as follows. Let i and $i+n$ be two copies of vertex i, where $p_{i+n} = p_i$ and $d_{i+n} = 0$. The model allows two service possibilities for each customer i. The pickup and delivery operations may be performed simultaneously, in which case vertex i is visited and vertex $i+n$ is not visited. Otherwise, customer i is visited twice: delivery is made at vertex i and pickup at vertex $i+n$. We

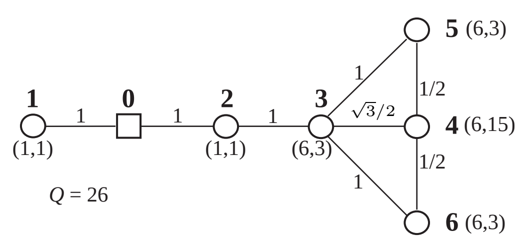


Figure 3: Euclidean instance for which the non-lasso solution $(0, 1, 2, 3, 5, 4, 6, 3, 0)$ is optimal. The pickup and delivery demands are indicated by (p_i, d_i).

define an extended cost matrix $\bar{C} = (\bar{c}_{ij})_{(2n+1)\times(2n+1)}$ where

$$\bar{c}_{ij} = \begin{cases} c_{ij} & \text{if } i \le n \text{ and } j \le n \\ c_{i-n,j} & \text{if } i > n \text{ and } j \le n \\ c_{i,j-n} & \text{if } i \le n \text{ and } j > n \\ c_{i-n,j-n} & \text{if } i > n \text{ and } j > n. \end{cases}$$

We also define the following variables:

$$x_{ij} = \begin{cases} 1, & \text{if the vehicle travels directly from } i \text{ to } j(i,j = 0,\ldots,2n; \\ & \quad i \ne j; j \ne i+n \text{ if } 1 \le i \le n; j \ne i-n \text{ if } i > n) \\ 0, & \text{otherwise.} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if pickup and delivery are performed simultaneously} \\ & \quad \text{at customer } i(i = 1,...,n) \\ 0, & \text{otherwise.} \end{cases}$$

$w_i =$ an upper bound on the total pickup amount in the vehicle upon leaving vertex $i(i = 0,\ldots,2n)$

$z_i =$ an upper bound on the total delivery amount in the vehicle upon leaving vertex $i(i = 0,\ldots,2n)$.

The general SVRPPD model is to

$$\text{minimize } \sum_{i=0}^{2n}\sum_{j=0}^{2n} \bar{c}_{ij}x_{ij} \tag{1}$$

subject to

$$\sum_{j=0}^{2n} x_{ij} = 1 \qquad (i = 0,\ldots,n) \tag{2}$$

$$\sum_{i=0}^{2n} x_{ij} = 1 \qquad (j = 0,\ldots,n) \tag{3}$$

$$\sum_{j=0}^{2n} x_{ij} = 1 - y_{i-n} \qquad (i = n+1,\ldots,2n) \tag{4}$$

$$\sum_{i=0}^{2n} x_{ij} = 1 - y_{j-n} \qquad (j = n+1,\ldots,2n) \tag{5}$$

$$w_0 = 0 \tag{6}$$

$$z_0 = \sum_{i=1}^{n} d_i \tag{7}$$

$$0 \leq w_i + z_i \leq Q \qquad (i = 1, \ldots, 2n) \tag{8}$$

$$w_j \geq w_i + p_j y_j - (1 - x_{ij})Q \qquad (i = 0, \ldots, 2n; j = 1, \ldots, n) \tag{9}$$

$$w_j \geq w_i + p_j(1 - y_{j-n}) - (1 - x_{ij})Q \qquad (i = 0, \ldots, 2n; j = n+1, \ldots, 2n) \tag{10}$$

$$z_j \geq z_i - d_j - (1 - x_{ij})Q \qquad (i = 0, \ldots, 2n; j = 1, \ldots, n) \tag{11}$$

$$x_{ij} \in \{0, 1\} \qquad (i, j = 0, \ldots, 2n, i \neq j; j \neq i + n \text{ if } 1 \leq i \leq n; j \neq i - n \text{ if } i > n) \tag{12}$$

$$y_i \in \{0, 1\} \qquad (i = 1, \ldots, n). \tag{13}$$

In this formulation, constraints (2) and (3) mean that the first vertex associated with each customer is visited once, either for a single delivery or for a simultaneous pickup and delivery. Constraints (4) and (5) express the fact that the second vertex associated with a customer is visited only if a combined pickup and delivery does not occur at the first vertex. Constraints (6) and (7) define the vehicle load upon leaving the depot, while constraints (8) guarantee that the vehicle load will never exceed the vehicle capacity. Constraints (9) and (10) state that the pickup amount in the vehicle is increased by p_j if vertex j is visited immediately after vertex i and a pickup takes place at that vertex. Constraints (11) mean that the delivery amount in the vehicle decreases by d_j if vertex j is visited immediately after vertex i. Constraints (12) and (13) impose the binary conditions on the variables. As in Desrochers and Laporte (1991), constraints (11) prevent the formation of subtours.

The size of instances that can be solved optimally using this model is relatively small and heuristics must therefore be used on practice. One such heuristic is an adaptation of the Unified Tabu Search Heuristic (UTSA) of Cordeau, Laporte and Mercier (2001). This heuristic has proved to be highly efficient on a host of vehicle routing problems and it easily adapts to several situations. The main features of the tabu search algorithm for the general SVPDP can be summarized as follows.

Initial solution UTSA is essentially an improvement procedure which can be applied to any solution, but as Gribkovskaia et al. (2006) have shown, it can pay to start from a good solution. These authors have developed construction heuristics specialized to the general SVPDP with combined demands which yields an excellent performance of the tabu search post-optimizer. One of the best construction heuristics can be summarized as follows. First construct a Hamiltonian undirected cycle $(0, i_1, \ldots, i_n, 0)$ by means of a TSP constructive heuristic, and consider the two directed circuits $(0, i_1, \ldots, i_n, 0)$ and $(0, i_n, \ldots, i_1, 0)$. From the first circuit, derive several solutions by removing one arc: for the first circuit 1) remove arc $(0, i_1)$ and construct the double-path solution $(0, i_n, \ldots, i_1, i_2, \ldots, i_n, 0)$; 2) remove arc (i_t, i_{t+1}) where $1 \leq t \leq n-1$, and construct the solution $(0, i_1, \ldots, i_t, i_{t-1}, \ldots, i_1, i_n, i_{n-1}, \ldots, i_{t+1}, i_{t+2}, \ldots, i_n, 0)$. Only the first of these solutions is guaranteed to be feasible for the SVPDP. Proceed similarly for the second circuit. Note that all these solutions contain one or two double-paths. Then apply the following *backward merging procedure* to each solution. Starting with the penultimate vertex of a double-path, combine the two visits made at that vertex into a single visit if this is feasible. Proceed to the previous vertex of the double-path, moving towards the depot, and continue until all vertices of the double-path have been scanned. Then go to the penultimate vertex of the other double-path if there are two double-paths, and continue until all vertices have been considered. Select the overall best feasible solution.

Penalized objective function In order to allow a mix of feasible and infeasible solutions, the algorithm works with a penalized function $f(s) = c(s) + \alpha q(s)$, where $c(s)$ is the cost of solution s, $q(s)$ is the total vehicle capacity violation of s, and α is a self-adjusting parameter. At each iteration, α is divided by $1 + \delta > 1$ if the current solution is feasible, and multiplied by $1 + \delta$ otherwise, where δ is a user-controlled parameter. This mechanism is identical to that of Cordeau, Gendreau and Laporte (1997).

Neighbourhood structure and attribute set At each iteration, the neighbourhood $N(s)$ of a solution s is defined as the set of all solutions reachable from s by changing the number v of visits of a customer. With s is associated an attribute set $B(s) = \{(i, v) : i \in V\backslash\{0\}$ and $v = 1$ or $2\}$. A transition from s to a neighbour s' is called a *move*, which can be defined as the removal of an attribute (i, v) from $B(s)$ and the inclusion of (i, v') in $B(s')$, where $v' \neq v$. There are two possible moves.

1. If $v = 1$, a second visit of i is inserted in the current solution so as to yield the smallest increase of $f(s)$. Visiting i twice will typically increase $c(s)$ and decrease $q(s)$.
2. If $v = 2$, the second occurrence of vertex i in the solution is deleted and its predecessor and successor are linked together. As a result pickup and delivery are now performed simultaneously at vertex i. Visiting i once will typically decrease $c(s)$ and increase $q(s)$.

Tabu status of an attribute If (i, v) is replaced with (i, v'), then the reinclusion of (i, v) in $B(s)$ is forbidden for θ iterations, where θ is a user-controlled parameter.

Aspiration criterion The *aspiration level* σ_{iv} of attribute (i, v) is initially defined as the cost of the initial solution s if $(i, v) \in B(s)$ and s is feasible; otherwise, $\sigma_{iv} = \infty$. Every time a feasible solution is identified, the aspiration level of attribute (i, v) is updated to $\min\{\sigma_{iv}, c(s)\}$. When considering a solution s' obtained by the inclusion of a tabu attribute (i, v) in $B(s')$, the tabu status of (i, v) is revoked if $q(s') = 0$ and $c(s') < \sigma_{iv}$. Let $M(s) \subseteq N(s)$ be the subset of neighbour solutions satisfying the aspiration criterion, i.e. solutions for which $(i, v') \in B(s')\backslash B(s)$ and s' is non-tabu, or s' is feasible and $c(s') < \sigma_{iv}$. In other words, the search only moves to a tabu solution if this yields a new best solution among those possessing attribute (i, v).

Diversification In order to diversity the search, any solution $s' \in M(s)$ such that $f(s') \geq f(s)$ is penalized by a term $p(s') = \zeta\rho_{iv}c(s)$, where ζ is a positive parameter, and ρ_{iv} is the relative frequency of all iterations for which attribute (i, v) has been in $B(s')$; if $f(s') < f(s)$, then $p(s') = 0$. The best $s' \in M(s)$ is the solution for which $g(s') = f(s') + p(s')$ is minimized. This type of diversification scheme was introduced by Glover (1989) and was later fine tuned by Taillard (1993).

Local reoptimization Route reoptimization is performed every φ iterations, where φ is a user-controlled parameter, or whenever a new best feasible solution is encountered. This is done by means of an improvement heuristic (Gribkovskaia et al., 2006) in which the penalized function $f(s)$ is used instead of $c(s)$.

Termination criterion The algorithm is applied for a fixed number η of iterations, where η is a user-controlled parameter.

The algorithm just described was extensively tested by Gribkovskaia et al. (2006) on benchmark instances derived from VRPLIB (http://www.er.deis.unibo.it/research_pages/ORinstances/VRPLIB/VRPLIB.html) containing between 16 and 101 vertices. It was observed that 38% of all solutions were non-Hamiltonian and 18% were non-lasso. This shows that it would have been suboptimal to impose a predefined shape on the solution. The frequency of multiple visits is higher in instances containing customers located close to the depot and having a large pickup demand compared with their delivery demand, so that it is preferable to perform the delivery and pickup operations separately.

4 Extensions of the general SVPDP

There exist several natural and meaningful extensions of the 1-M-1 general SDVRP, all of which have only received limited attention. We will consider four such extensions. In these extensions, visiting customers twice may be dictated by feasibility considerations.

4.1 Periodic SVPDPs

In periodic problems customer pickup and delivery requirements are spread over a period of several days and the problem is then to simultaneously determine a subset of customers and the order of visits for each day. This problem is encountered in the planning of reverse logistics operations, for example when new household appliances and furniture such as washing machines, fridges or mattresses must be delivered and used items must be collected. Alshamrani, Mathur and Ballou (2007) have studied the case where the pickup and delivery operations may be spread over several days but a maximum time limit is imposed between the pickup and the delivery operations in order to avoid product deterioration. The problem studied by these authors is motivated by the blood distribution system of the American Red Cross.

4.2 SVPDPs with selective pickups

An interesting case, also arising in reverse logistics, is where pickups are optional, but generate a profit when performed. An example described by Privé et al. (2006) is the delivery of beverages to supermarkets and convenience stores, and the collection of empty recyclable containers. This case was recently studied by Gribkovskaia, Laporte and Shyshou (2006). To handle this variant, the authors introduce an additional binary variable $u_i (i = 1, \ldots, n)$ to the model of Section 3.2, equal to 1 if and only if pickup is performed during the second visit to customer i. If the pickup associated with customer i generates a revenue r_i, then the objective becomes

$$\text{minimize} \sum_{i=0}^{2n} \sum_{j=0}^{2n} \bar{c}_{ij} x_{ij} - \sum_{i=1}^{n} r_i y_i - \sum_{j=n+1}^{2n} r_{j-n} u_{j-n}. \tag{14}$$

The right-hand sides of constraints (4) and (5) become u_{i-n} and u_{j-n}, respectively. It is also necessary to impose the constraints

$$u_i + y_i \leq 1 \qquad (i = 1, \ldots, n) \tag{15}$$

and to modify constraints (10) as follows:

$$w_j \geq w_i + p_{j-n} u_{j-n} - (1 - x_{ij})Q \qquad (i = 0, \ldots, 2n; j = n+1, \ldots, 2n). \tag{16}$$

In the tabu search algorithm for this problem, the neighbourhood structure is similar to that of the general SVPDP, but a status is assigned to each customer: PD for a simultaneous pickup and delivery, D for a single delivery, and P-D for a separate pickup and delivery operations. Then,

1) if $v = 1$, a second visit of i is inserted in the current solution so as to yield the smallest increase of $f(s)$. In other words, for each vertex with status PD or D change to status P-D is evaluated;
2) if $v = 2$, the second occurrence of vertex i in the solution is deleted and its predecessor and successor are connected. Or, in terms of vertex statuses, for each vertex with status P-D two possible modifications to the status PD or D are evaluated and only the modification yielding the smaller increase of $f(s)$ is considered for each vertex. As a result either pickup and delivery are performed simultaneously at vertex i or only delivery demand is satisfied.

In addition local reoptimization is applied whenever a new best feasible solution is identified. This is done by means of two improvement heuristics, called SP and RC, which are used every ψ^{th} and φ^{th} iteration, respectively, where ψ and φ are user-controlled parameters. Heuristic SP (shifting pickups) attempts to improve the solution by delaying the pickup operation of PD customers, and thus freeing some space on the vehicle; heuristic

RC (resequencing of customers) attempts to improve the solution by iteratively removing from the route vertices visited once and reinserting them in the most profitable position.

Several variants of this algorithm were extensively tested. One of the best two yielded solutions within 3.98% of the minimal reachable routing cost, and 0.24% of the optimal revenue; for the other one, these figures were 3.35% and 0.55%.

Finally, we mention that Süral and Bookbinder (2003) have also formulated and solved a version of the SVPDP with selective pickups in which each customer has a pickup or a delivery but not both.

4.3 SVPDPs with intermediate drops

Another variant of the general SVPDP with combined demands is to allow drops at intermediate vertices. For example, when the vehicle makes several passages through the depot, it may make sense to empty part of its content in order to create extra space and thus allow more flexibility in the remaining part of the route. In the example of Figure 4, an optimal solution of cost 9 is (0,1,2,3,5,4,6,3,0) if no intermediate drop at the depot is allowed. However, a better solution (0,1,0,2,3,5,4,6,0) of cost 8.91 is obtained if the pickup demand of customer 1 ($p_1 = 6$) is dropped at the depot while the vehicle is traveling from 1 to 2. Another possibility is to allow transshipment at customer locations along the vehicle route. Consider the example of Figure 5 in which edge (0,1) is rotated clockwise so that the cost of edge (1,2) becomes 1.97. Then an optimal solution without intermediate drop is still (0,1,2,3,5,4,6,3,0) and has a cost of 8.97. Transshipping six demand units at vertex 2 yields a better solution (0,1,2,3,5,4,6,2,0) of cost 8.90. As far as the authors are aware, intermediate drops have never been studied in the context of the SVPDP, but Mitrović-Minić and Laporte (2006) have shown the benefits of transshipment in the context of 1-1 pickup and delivery problems.

4.4 SVPDPs with capacitated customers

Finally, another interesting extension of the general SVPDP is the case of capacitated customers. In some contexts like the servicing of offshore platforms (Gribkovskaia, Laporte

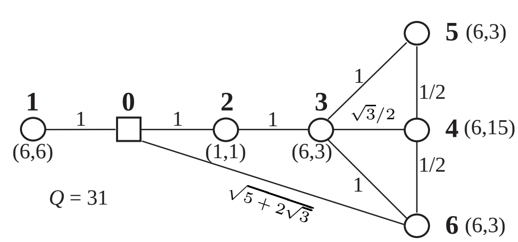


Figure 4: Euclidean instance for which an intermediate drop at the depot is beneficial. The pickup and delivery demands are indicated by (p_i, d_i).

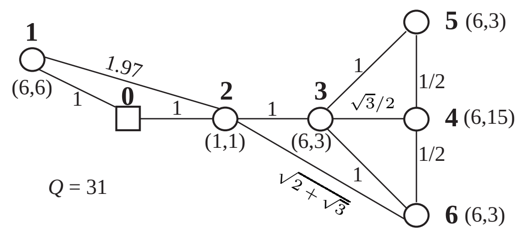


Figure 5: Euclidean instance for which transshipment at a customer location is beneficial. The pickup and delivery demands are indicated by (p_i, d_i).

and Shlopak, 2006), there is no spare capacity at the customer locations. When a vehicle (a vessel) arrives at a platform, it may have to first unload a container from the platform before performing a delivery, but this is only possible if there is sufficient capacity in the vessel. This situation is handled by adding the following constraints to the model of Section 3.2. Let C_i be the available free capacity of customer i at the start of operations (it is assumed that $C_i \geq d_i - p_i$ to ensure feasibility). Then

$$C_i \geq d_i - p_i y_i \qquad (i = 1, \dots, n) \tag{17}$$

and

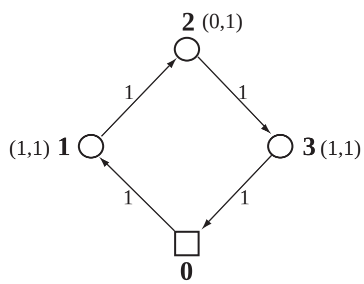
$$(Q - w_i - z_i) + C_i \geq 1 \qquad (i = 1, \dots, n) \tag{18}$$

Constraints (17) ensure that there is sufficient capacity at each customer location to perform the pickup and delivery services, while constraints (18) prevent infeasible situations in which the vehicle would arrive fully laden at a location with no free storage space, and the amounts to be picked up and delivered would be the same. More specifically, these constraints state that the amount of free space on the vehicle and at the customer location cannot both be zero.

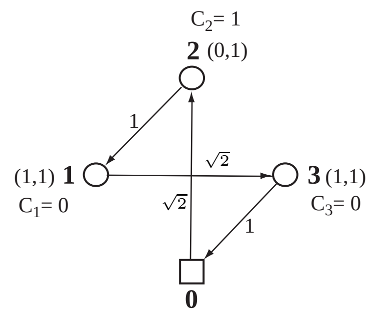
Figure 6 depicts a case where $Q = 3$. In Figure 6a there are no customer capacities and the solution $(0, 1, 2, 3, 0)$ of cost 4 is optimal. In Figure 6b customer capacities are imposed and the Hamiltonian solution $(0, 2, 1, 3, 1)$ of cost $2+2\sqrt{2}$ is optimal. In Figure 6c no Hamiltonian solution is feasible and the non-Hamiltonian solution $(0, 2, 1, 2, 3, 0)$ of cost $4 + \sqrt{2}$ is optimal.

In order to handle customer capacities, some modifications must be made to the tabu search algorithm of Section 3.2. Vertices are first classified into three categories:

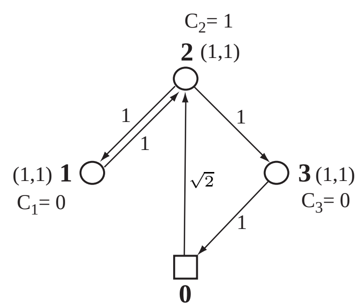
category 0: vertices for which $C_i = 0$ and $d_i = p_i$;
category 1: vertices for which $C_i < d_i$;
category 2: vertices for which $C_i \geq d_i$.



a) Optimal solution without customer capacities



b) Optimal Hamiltonian solution with customer capacities



c) Optimal non-Hamiltonian solution with customer capacities

Figure 6: Instance with capacitated customers. The vehicle capacity is $Q = 3$. The pickup and delivery demands are indicated by (p_i, d_i).

Vertices of category 0 and 1 can only be visited once for a simultaneous pickup and delivery, and those of category 0 can only be visited when the vehicle is not fully laden. This is obvious because these vertices do not have sufficient available capacity to accept their delivery demand without their pickup demand being collected. Vertices of category 2 can be visited once or twice on a route.

Moreover, a solution is said to be *load-feasible* if the vehicle capacity is never exceeded. It is *storage-feasible* if none of the vertices of categories 0 or 1 is visited twice. It is *operational-feasible* if a fully laden vehicle never serves a customer with no available capacity (category 0 vertex). Then the following changes are implemented in the search procedure.

Neighborhood *N(s)* and definition of a move The neighborhood $N(s)$ of solution s is defined by all solutions that can be reached from s by changing the number of visits at one category 2 vertex.

Load feasibility violations Load feasibility is checked whenever a vertex is visited. The total load infeasibility of a route is equal to the sum of load infeasibilities of all its vertices.

Operational feasibility violations Operational feasibility means that vertices with zero capacity must not be visited by a fully laden vehicle. The operational feasibility violation of a route is the number of such vertices.

Storage feasibility violations Vertex capacity violations are not allowed. Before solving an instance, vertices that can be visited only once (the number of visits is not dependent on routing) are identified.

Penalized objective function For a solution $s \in S$, let $c(s)$ denote the total routing cost, let $q(s)$ denote the total load violation of the route, and let $g(s)$ denote the operational feasibility violation of the route. Solutions $s \in S$ are evaluated with the help of the penalized cost function $f(s) = c(s) + \alpha q(s) + \pi g(s)$, where α and π are positive self-adjusting parameters.

Tests have shown that this modified tabu search algorithm can effectively solve realistic instances and the best found solutions are not always Hamiltonian.

5 Conclusions

The one-to-many-to-one SVPDP arises in several practical contexts and has been extensively studied by operations researchers. Several variants of the problem exist according to whether pickup and delivery operations can or must be performed separately, and to whether an order is imposed on the sequencing of these two types of operation. Allowing general solutions in which no a priori shape is imposed and customers may be visited once or twice is often beneficial. Because these problems are rather hard to solve, very few exact algorithms are available. More often than not, heuristics are the only practical solution methodology. For some variants of the problem, heuristics with a bounded worst case performance ratio have been proposed. Several variants can be solved efficiently by modifying the Unified Tabu Search Algorithm of Cordeau, Laporte and Mercier (2001). Various extensions of the 1-M-1 SVPDP have been described. Some of these are relevant to the planning of reverse logistics operations and to the servicing of offshore oil and gas platforms.

References

Alshamrani, A., Mathur, K., and Ballou, R.H. (2007). Reverse logistics: Simultaneous design of delivery routes and return strategies. *Computers & Operations Research*, 34:595–619.

Anily, S. and Hassin R. (1992). The swapping problem. *Networks*, 22:419–433.

Anily, S. and Mosheiov, G. (1994). The traveling salesman problem with delivery and backhauls. *Operations Research Letters*, 16:11–18.

Baldacci, R., Hadjiconstantinou, E.A., and Mingozzi, A. (2003). An exact algorithm for the traveling salesman problem with deliveries and collections. *Networks*, 42:26–41.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: A classification scheme and survey. *TOP*, forthcoming.

Chisman, J.A. (1975). The clustered traveling salesman problem. *Computers & Operations Research*, 2:115–118.

Christofides, N. (1976). Worst-case analysis of a new heuristic for the travelling salesman problem. Research report 388, G.S.I.A., Carnegie Mellon University, Pittsburgh, PA.

Cordeau, J.-F., Gendreau, M., and Laporte, G. (1997). A tabu search algorithm for periodic and multi-depot vehicle routing problems. *Networks*, 30:105–119.

Cordeau, J.-F., Laporte, G., and Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52:928–936.

Desrochers, M. and Laporte, G. (1991). Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10:27–36.

Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. *Operations Research Spectrum*, 23:79–96.

Frederickson, G., Hecht, M., and Kim, C. (1978). Approximation algorithms for some routing problems. *SIAM Journal on Computing*, 7:178–193.

Gendreau, M., Hertz, A., and Laporte, G. (1997). An approximation algorithm for the traveling salesman problem with backhauls. *Operations Research*, 45:639-641.

Gendreau, M., Laporte, G., and Vigo, D. (1999). Heuristics for the traveling salesman problem with pickup and delivery. *Computers & Operations Research*, 26:699-714.

Glover, F. (1989). Tabu search – Part 1. *ORSA Journal on Computing*, 1:190–206.

Golden, B.L., Bodin, L.D., Doyle, T., and Stewart, W.R. Jr. (1980). Approximate travelling salesman algorithms. *Operations Research*, 28:694–711.

Gribkovskaia, I., Halskau sr., Ø., Laporte, G., and Vlček, M. (2006). General solutions to the single vehicle routing problem with pickups and deliveries. *European Journal of Operational Research*, forthcoming.

Gribkovskaia, I., Laporte, G., and Shlopak, A. (2006). A tabu search heuristic for a routing problem arising in the servicing of offshore platforms. Submitted for publication.

Gribkovskaia, I., Laporte, G., and Shyshou, A. (2006). The single vehicle routing problem with deliveries and selective pickups. Submitted for publication.

Mitrović-Minić, S. and Laporte G. (2006). The pickup and delivery problem with time windows and transshipment. *INFOR*, 44:217–227.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24:1097–1100.

Mosheiov, G. (1994). The travelling salesman problem with pick-up and delivery. *European Journal of Operational Research*, 79:299–310.

Nagy, G., and Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, 162:126–141.

Privé, J., Renaud, J., Boctor, F.F., and Laporte, G. (2006). Solving a vehicle routing problem arising in soft drink distribution. *Journal of the Operational Research Society*, 57:1045–1052.

Süral, H. and Bookbinder, J.H. (2003). The single-vehicle routing problem with unrestricted backhauls. *Networks*, 41:127–136.

Taillard, É.D. (1993). Parallel iterative search for vehicle routing problems. *Networks*, 23:661–673.

Tang, F.A. and Galvão, R.D. (2002). Vehicle routing problems with simultaneous pick-up and delivery service. *Opsearch*, 39:19–33.

Tang, F.A. and Galvão, R.D. (2006). A tabu search algorithm for the vehicle routing probem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33:595–619.

Wasner, M. and Zäphel, G. (2004). An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. *International Journal of Production Economics*, 90:403–419.