

# Static pickup and delivery problems: a classification scheme and survey

Gerardo Berbeglia · Jean-François Cordeau ·  
Irina Gribkovskaia · Gilbert Laporte

© Sociedad de Estadística e Investigación Operativa 2007

**Abstract** Pickup and delivery problems constitute an important class of vehicle routing problems in which objects or people have to be collected and distributed. This paper introduces a general framework to model a large collection of pickup and delivery problems, as well as a three-field classification scheme for these problems. It surveys the methods used for solving them.

**Keywords** Vehicle routing · Stacker crane · Swapping problem · Backhauls · Dial-a-ride problem

**Mathematics Subject Classification (2000)** 90-02 · 90B06

---

This invited paper is discussed in the comments available at:

<http://dx.doi.org/10.1007/s11750-007-0010-7>, <http://dx.doi.org/10.1007/s11750-007-0011-6>,  
<http://dx.doi.org/10.1007/s11750-007-0012-5>, <http://dx.doi.org/10.1007/s11750-007-0013-4>,  
<http://dx.doi.org/10.1007/s11750-007-0014-3>.

---

G. Berbeglia (✉) · G. Laporte  
Canada Research Chair in Distribution Management, HEC Montréal, 3000 chemin de la  
Côte-Sainte-Catherine, Montreal H3T 2A7, Canada  
e-mail: [gerardo.berbeglia@hec.ca](mailto:gerardo.berbeglia@hec.ca)

G. Laporte  
e-mail: [gilbert@crt.umontreal.ca](mailto:gilbert@crt.umontreal.ca)

J.-F. Cordeau  
Canada Research Chair in Logistics and Transportation, HEC Montréal, 3000 chemin de la  
Côte-Sainte-Catherine, Montreal H3T 2A7, Canada  
e-mail: [cordeau@crt.umontreal.ca](mailto:cordeau@crt.umontreal.ca)

I. Gribkovskaia  
Molde University College, Postbox 2110, 6402 Molde, Norway  
e-mail: [irina.gribkovskaia@himolde.no](mailto:irina.gribkovskaia@himolde.no)

## 1 Introduction

*Pickup and delivery problems* (PDPs) constitute an important class of vehicle routing problems in which objects or people have to be transported between origins and destinations. These problems, which have been studied for more than 30 years, arise in many contexts such as logistics, ambulatory services, and robotics. Most existing PDPs can be defined within the following general framework. Let  $G = (V, A)$  be a complete and directed graph with vertex set  $V = \{0, \dots, n\}$ , where vertex 0 represents the depot, and each remaining vertex represents a customer. The arc set is defined as  $A = \{(i, j) : i, j \in V, i \neq j\}$ . Each arc  $(i, j) \in A$  has a non-negative length or cost  $c_{ij}$  (usually equal to the travel time) satisfying the triangle inequality. Let  $H = \{1, \dots, p\}$  be a set of commodities representing types of entities to be transported. Each vertex, including the depot, can either need or supply a non-negative amount of each commodity. Let  $D = (d_{ih})$  denote a *commodity matrix*, where a positive  $d_{ih}$  is the amount of commodity  $h$  supplied by vertex  $i$ , and  $-d_{ih}$  is the amount of commodity  $h$  required by vertex  $i$  if  $d_{ih} < 0$ . In some contexts associated with passenger transportation, it is more natural to use the term “request” instead of “commodity”, but from a modeling point of view these two terms can be used interchangeably. It is assumed that  $\sum_{i \in V} d_{ih} = 0$  for each commodity  $h \in H$ . That is, for each commodity the total supply and the total demand are in equilibrium. Let  $K = \{1, \dots, m\}$  be a set of available vehicles, each of capacity  $Q$ . The subset  $T \subseteq V$  designates transshipment vertices at which vehicles are allowed to temporarily drop an amount of any commodity and pick it up later, possibly with another vehicle. A vehicle can either pick up or deliver the entire amount of a commodity available or required at a vertex. A *route* is a circuit over some vertices, starting and finishing at the depot. PDPs consist of constructing at most  $m$  vehicle routes such that:

- (i) all pickup and delivery requests are satisfied;
- (ii) no transshipments of commodities are made at vertices of  $V \setminus T$ ;
- (iii) the load of a vehicle never exceeds its capacity;
- (iv) the sum of route costs is minimized.

Since usually the cost  $f$  of a route consists of the sum of the costs associated with the arcs used by the route, this will be assumed unless stated otherwise. Additional assumptions and constraints on vehicle routes may be present depending on a specific problem, such as time windows, or multiple visits at some vertices. In some problems with specific side constraints, the cost function  $f$  may include additional information such as the quality of service to the users.

It is convenient to classify PDPs according to the following simple three-field scheme: [Structure|Visits|Vehicles]. The first field, called *structure*, specifies the number of origins and destinations of the commodities. In *many-to-many* problems (M-M), any vertex can serve as a source or as a destination for any commodity. In *one-to-many-to-one* problems (1-M-1), commodities are initially available at the depot and are destined to the customer vertices; in addition, commodities available at the customers are destined to the depot. The sets of pickup and delivery customers are not necessarily disjoint (Ropke and Pisinger 2006b) and often coincide, as is the case in the distribution of beverages and the collection of empty cans and bottles (see, e.g., Privé et al. 2006). Finally, in *one-to-one* problems (1-1), each commodity has a given

origin and a given destination. Problems of this type arise, for example, in courier operations and door-to-door transportation services (see, e.g., Cordeau and Laporte 2003a). The second field provides information on the way pickup and delivery operations are performed at customer vertices. We use the notation PD to indicate that each customer vertex is visited exactly once for a combined pickup and delivery operation, P-D if these two operations may be performed together or separately, and P/D for the case where each customer vertex either has a pickup or a delivery request but not both, so that either a pickup or a delivery operation is performed at each vertex. In addition, we use the letter T whenever some vertices may be used as transshipment points. Finally, the third field gives the number of vehicles used in the solution. Whenever a field is undefined, we use the notation “—”. Clearly, this classification does not fully capture all intricacies of PDPs, such as side constraints, alternative objectives, number of commodities, etc., but we believe such schemes only work if they are parsimonious.

The different origin-destination structures and the varied number of commodities of PDPs are also reflected in the structure of the commodity matrix. For instance, in a single-commodity problem known as the *Q-Delivery Traveling Salesman Problem* (Q-DTSP) (Anily and Bramel 1999), the commodity matrix  $D$  reduces to a vector. In the *Dial-a-Ride Problem* (DARP) (Cordeau and Laporte 2003a), the commodity matrix contains exactly one non-zero value in each row other than the one associated with the depot. This framework also allows the modeling of PDPs with transshipments, where objects may be dropped temporarily at intermediate vertices (Mitrović-Minić and Laporte 2006). In this case, the rows in the commodity matrix associated to the transfer vertices contain only zero entries. An advantage of this framework over that of Savelsbergh and Sol (1995) is that it allows modeling many-to-many problems such as the Q-DTSP, the *Swapping Problem* (SP) (Anily and Hassin 1992), and the *One-Commodity Pickup and Delivery Traveling Salesman Problem* (1-PDTSP) (Hernández-Pérez and Salazar-González 2004b).

Another important dimension present in PDPs relates to the availability of information. In *static* problems, all information is assumed to be deterministic and known a priori. In *dynamic* problems, information is gradually revealed over time. Solving dynamic problems entails devising a solution strategy that will adjust a current solution in the light of new information (Mitrović-Minić et al. 2004). In *stochastic* problems, some of the data (e.g., demands, travel times, etc.) are random variables whose distributions are usually known (Powell et al. 2005; Fu and Rilett 1998). This paper is devoted to static problems which have been studied the most. The remainder of this paper is organized as follows. Sections 2, 3, and 4 are devoted to the three important classes of PDPs called many-to-many problems, one-to-many-to-one problems, and one-to-one problems, respectively. Conclusions follow in Sect. 5.

## 2 Many-to-many problems

Pickup and delivery problems with a many-to-many structure are characterized by several origins and destinations for each commodity. The relevant literature refers mostly to single vehicle scenarios and is rather limited. This can probably be explained by the fact that many-to-many problems are not frequently encountered in

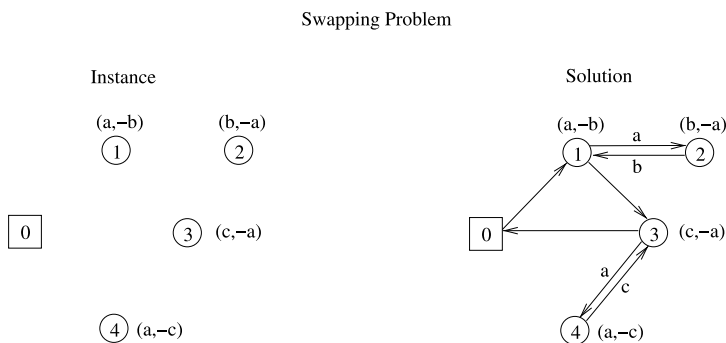
practice. Three problems will be reviewed in this section. Sections 2.1, 2.2, and 2.3 survey the literature of the *Swapping Problem*, the *One-Commodity Pickup and Delivery Traveling Salesman Problem*, and the *Q-Delivery Problem*, respectively. The first of these problems is a multi-commodity problem, while the other two deal with a single commodity.

## 2.1 The swapping problem [M-M|P-D|1]

The *Swapping Problem* (SP) is a many-to-many,  $n$ -commodity PDP introduced by Anily and Hassin (1992). It deals with the swapping of objects between vertices of  $V \setminus \{0\}$  with a single vehicle. The SP contains the following restrictions: (i)  $m = 1$ ; (ii)  $Q = 1$ ; (iii) the row associated to each vertex  $j \in V$  in the matrix  $D$  has at most two non-zero values  $d_{jh_1}$  and  $d_{jh_2}$  in the set  $\{-1, 1\}$  with  $d_{jh_1} \neq d_{jh_2}$ . The first and second restrictions state that the SP is a single vehicle problem with unit capacity. The third restriction states that each vertex requests or supplies at most one commodity. In addition, the amount requested or supplied is one unit. Figure 1 depicts a simple example with three commodities and four customer vertices.

The SP can be preemptive or non-preemptive. In the preemptive case  $T = V$  means that objects may be temporarily dropped at any vertex. This case is denoted by [M-M|P-D-T|1]. In the non-preemptive case no transshipments are allowed, i.e.,  $T = \emptyset$ . The version of the SP in which  $T \neq \emptyset$  and  $T \neq V$  is sometimes called the *Mixed Swapping Problem*. The restriction that each vertex supplies or demands at most one object can be overcome by replicating vertices that supply or demand multiple objects. As noted by Anily and Hassin (1992), the SP is NP-hard (Garey and Johnson 1979), even for a unit capacity vehicle. To see this, consider an instance of the *Traveling Salesman Problem* (TSP), duplicate each vertex  $i$  into  $i'$  and  $i''$ , set  $d_{i'1} = d_{i''2} = 1$ ,  $d_{i'2} = d_{i''1} = -1$ ,  $c_{i'i''} = 0$ ,  $H = \{1, 2\}$ ,  $T = \emptyset$  and set  $c_{x'y'} = c_{x'y''} = c_{x''y'} = c_{x''y''} = c_{xy}$  for every edge  $(x, y)$ . Then any optimal solution of the SP will result in an optimal solution of the TSP instance.

Anily and Hassin (1992) have characterized the structure of optimal solutions and have developed two heuristics with a worst-case performance ratio of 2.5. These heuristics are based on the solution of a relaxation of the problem which is then



**Fig. 1** Example of a non-preemptive Swapping Problem instance with vertices  $\{0, 1, 2, 3, 4\}$  and commodities  $\{a, b, c\}$ . Vertex labels  $(x, -y)$  mean that the vertex supplies  $x$  and demands  $y$ . Arc labels denote the commodity carried by the vehicle

extended to a feasible solution in such a way that the cost is increased by a constant factor. The relaxation consists of the solution of  $m + 1$  minimum cost assignment problems on a complete bipartite graph, one for each object type. A set of disjoint cycles is then obtained and patched into a single Eulerian tour. Chalasani and Motwani (1999) have derived a better bound with a worst-case ratio performance of 2 for a special case of the SP with two object types, called the *Bipartite Traveling Salesman Problem*. The algorithm is based on a matroid intersection procedure (Oxley 1993).

Anily et al. (1999) have studied the SP, where vertices are located on a line. The authors have derived some general properties of optimal solutions and have developed an exact  $O(n^2)$  algorithm for the problem. More recently, the same authors (Anily et al. 2006) have considered the preemptive swapping problem on a tree. They have shown that the problem is NP-hard, by reduction of the *Steiner Tree Problem* on a bipartite graph (Garey and Johnson 1979). They have also presented a heuristic with a worst-case performance ratio of 1.5, and they have shown that in the single commodity case the problem can be solved in polynomial time. A linear time algorithm for the single-commodity, a non-preemptive version of the SP on a tree, is presented by Wang et al. (2006) (see Sect. 2.3).

## 2.2 The one-commodity pickup and delivery traveling salesman problem [M-M|P/D|1]

The *One-Commodity Pickup and Delivery Traveling Salesman Problem* (1-PDTSP) is a many-to-many, single-commodity problem. The set of customer locations is partitioned into pickup and delivery customers to be served by a single vehicle based at the depot. The 1-PDTSP contains the following restrictions: (i)  $m = 1$ ; (ii)  $p = 1$ ; (iii)  $T = \emptyset$ . The first and the second restriction state that the 1-PDTSP is a single vehicle, single-commodity problem. The third restriction ensures that the solution is Hamiltonian.

The 1-PDTSP, introduced by Hernández-Pérez and Salazar-González (2004a), is NP-hard and is closely related to a special case of the SP. Consider a non-preemptive SP in which multiple demands and supplies of objects are allowed (this can be done by vertex replications). Assume also that there is only one type of object provided or requested at each vertex. Then the 1-PDTSP is equivalent to this SP with the additional constraint that the tour must be Hamiltonian. In the 1-PDTSP, the load carried by a vehicle may vary along the route, which makes it difficult to find a feasible solution. In fact, deciding whether a 1-PDTSP instance is feasible or not is NP-complete, since the *3-Partitioning Problem* can be shown to be a special case of this problem (Hernández-Pérez and Salazar-González 2003).

Hernández-Pérez and Salazar-González (2004a) have developed a branch-and-cut algorithm for the 1-PDTSP, capable of solving instances with up to 40 vertices. The authors have proposed an integer linear programming (ILP) model equivalent to a classical TSP model with a set of Benders cuts. This implies that all known valid inequalities for the TSP are also valid for the 1-PDTSP. The authors have proposed a maximum flow algorithm and fast heuristics to identify violated Benders cuts, and they have used clique inequalities from the set-packing polytope (Balas and Padberg 1976).

Hernández-Pérez and Salazar-González (2004b) have later developed two heuristics for the 1-PDTSP and were able to solve instances with up to 500 vertices. Given a path  $P$  starting at the depot, the authors define an associated measure called the infeasibility of the path  $P$  in order to quantify how infeasible the path is with respect to the capacity constraints. The first heuristic uses a modified version of the TSP nearest neighbor algorithm. Travel costs are redefined in order to favor connections between pickup and delivery customers, making it more likely to generate a feasible solution. The other important ingredient of the heuristic is the adaptation of the classical 2-opt and 3-opt edge exchange heuristics (Lin 1965; Johnson and McGeoch 1997), where not all possible exchanges are considered. These exchanges are applied to minimize the infeasibility of the tour under some constraints. The heuristic iterates between a greedy algorithm and a tour improvement procedure. The second heuristic is an incomplete optimization algorithm that uses the exact methods developed by Hernández-Pérez and Salazar-González (2004a) to iteratively determine optimal solutions in restricted feasible regions.

### 2.3 The Q-delivery traveling salesman problem [M-M|P/D|1]

The Q-DTSP is a many-to-many, single-commodity PDP introduced by Chalasani and Motwani (1999). It consists of picking up and delivering single objects from source to sink vertices. The Q-DTSP contains the following restrictions: (i)  $m = 1$ ; (ii)  $p = 1$ ; (iii)  $d_{j1} \in \{1, -1\}$  for  $j \in V \setminus \{0\}$ ; (iv)  $d_{01} = 0$ . The first and second restriction state that the Q-DTSP is a single vehicle and single-commodity problem. The third condition specifies that each vertex, except the depot, supplies or demands one unit. The fact that the depot does not supply nor demand any unit is expressed by the fourth restriction. Since the Q-DTSP is a single-commodity problem, there is no advantage in temporarily dropping objects at some vertices. Thus, only one operation is performed at each vertex since each of them requires or supplies exactly one unit, and, therefore, the Q-DTSP is restricted to Hamiltonian tours. This fact, together with restrictions (i) and (ii), shows that the Q-DTSP is a special case of the 1-PDTSP. The Q-DTSP is NP-hard since it includes the TSP as a special case in which every delivery customer is paired with a pickup customer at the same location. A more general problem, called *General Q-Delivery Traveling Salesman Problem* (G-Q-DTSP), allows vertices to provide or demand any number of objects. The G-Q-DTSP is similar to the 1-PDTSP. The difference is that the tour is not necessarily Hamiltonian. For this reason it is sometimes seen as the *split version* of the 1-PDTSP. Motivated by problems appearing in industrial robotics, Chalasani and Motwani (1999) have studied the Q-DTSP as well as a new dynamic variant of the problem called the *Dynamic k-collect TSP*. They have described a polynomial approximation algorithm with a worst-case performance ratio of 9.5. It is based on the use of the 2.0 worst-case performance algorithm developed by the same authors for the unit-capacity version called the *Bipartite Traveling Salesman Problem*.

Anily and Bramel (1999) have studied the Q-DTSP under the name *Capacitated Traveling Salesman Problem with Pickups and Deliveries*. The authors have slightly modified the algorithm of Chalasani and Motwani (1999) and improved the worst-case bound from 9.5 to 7. In addition, they have derived a new polynomial-time algorithm in which the worst-case bound depends on the vehicle capacity  $Q$ . This new algorithm has a better worst-case performance than the previous one for all  $Q \leq 255$ .

The Q-DTSP on a line and on a tree has been studied by Wang et al. (2006). For the line case, the authors have presented an  $O(n^2/\min\{Q, n\})$  algorithm and have shown how to solve the unit capacity version ( $Q = 1$ ) and the uncapacitated version ( $Q = \infty$ ) in linear time. For the tree case, they have proved that the problem is NP-hard and they have presented an  $O(n)$  algorithm for the case in which  $Q = 1$ , as well as an  $O(n^2)$  algorithm for the case in which  $Q = \infty$ .

### 3 One-to-many-to-one pickup and delivery problems

In one-to-many-to-one pickup and delivery problems (1-M-1-PDPs), some commodities (called delivery commodities) are destined from the depot to customer vertices while other commodities (called pickup commodities) supplied by the customers are brought back to the depot. These problems arise, for instance, in reverse logistics where full containers must be brought to customers, and empty containers must be returned from the customers to the depot. Beullens et al. (2004) discuss several issues associated with such problems. Gribkovskaia et al. (2006) describe an application arising in the supply of offshore oil and gas platforms.

This class of problems contains the following restrictions: (i)  $p = 2$ ; (ii)  $d_{i1} \geq 0$ , for  $i \in V \setminus \{0\}$ ; (iii)  $d_{i2} \leq 0$ , for  $i \in V \setminus \{0\}$ ; (iv)  $T = \emptyset$ . The first restriction states that these problems, from a modeling point of view, have two commodities: a pickup commodity and a delivery commodity. This is because all delivery demands can be viewed as single commodity whenever they share a common origin, and all pickup demands can also be viewed as a single commodity whenever they share a common destination. The second and third restriction state that every customer can have a pickup demand for the pickup commodity and a delivery demand for the delivery commodity. The fourth restriction disallows transshipments.

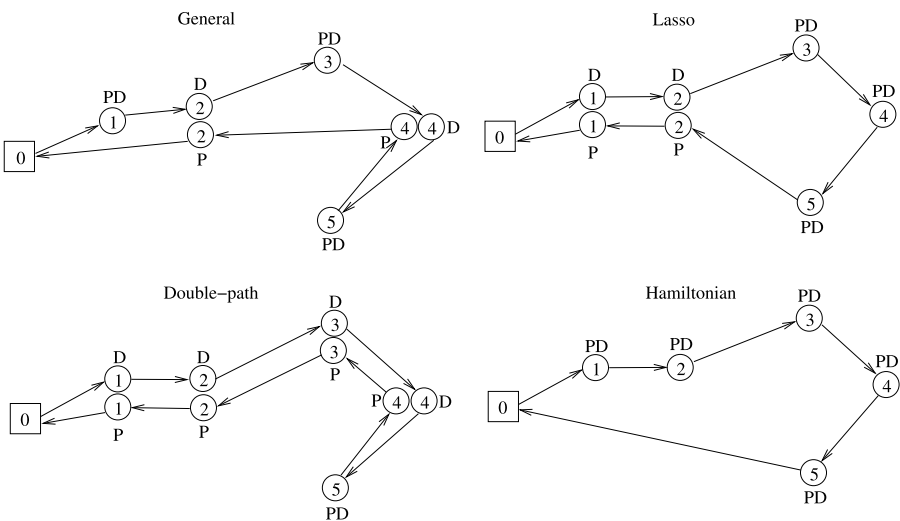
The remainder of this section consists of an overview of the different types of 1-M-1-PDPs and includes a review of the most relevant algorithms. Specific types of solutions arise from additional constraints on the routes. We will consider two main demand structures for 1-M-1-PDPs: *combined demands* and *single demands*. In problems with combined demands at least one customer has positive pickup and delivery demand. In this class of 1-M-1-PDPs, four types of solutions can be distinguished (see Sect. 3.1). In problems with single demands each customer has a zero pickup or delivery demand. In this class of 1-M-1-PDPs two solution types have been studied. A solution is said to be *mixed* if delivery and pickup customers can be served in any order. The problem of finding mixed solutions in this class is called the *1-M-1-PDP with Single Demands and Mixed Solutions*. On the other hand, the problem of finding solutions under the constraint that pickups can only be performed after all the deliveries have taken place is called the *1-M-1-PDP with Single Demands and Backhauls*.

The 1-M-1-PDP with Combined Demands and the 1-M-1-PDP with Single Demands induce different models and algorithms in some cases while in others both structures yield the same solution approaches. Sections 3.1 and 3.2 are devoted to these two classes of problems, respectively.

### 3.1 The 1-M-1-PDP with combined demands [1-M-1|P-D|–]

In the 1-M-1-PDP with Combined Demands there exists at least one vertex  $i$  ( $i \neq 0$ ), for which  $d_{i1}$  and  $d_{i2}$  are both non-zero. In their study of the *Single Vehicle 1-M-1-PDP with Combined Demands*, Gribkovskaia et al. (2007) identify four solution types: *general*, *Hamiltonian*, *lasso* and *double-path*. A general solution has no pre-determined shape: this means that any customer may be visited once or twice. A Hamiltonian solution is one in which each vertex is visited exactly once for a combined pickup and a delivery. In double-path solutions, each vertex is first visited along a path for a pickup, and then visited along a path to the depot for a delivery: this means that only the last vertex on the first path is visited once for a combined pickup and delivery. In a lasso solution, each vehicle follows a path performing deliveries, then visits the remaining customers along a loop by performing a simultaneous pickup and delivery at each visit, and finally a path is followed to perform the remaining pickups. Lasso solutions with an empty path are Hamiltonian, and lasso solutions with an empty loop are double-path solutions. Figure 2 illustrates various solution shapes. Let  $z_G^*$ ,  $z_L^*$ ,  $z_D^*$  and  $z_H^*$  be the values of general, lasso, double-path and Hamiltonian optimal solutions of the same instance, respectively. If the cost matrix satisfies the triangle inequality, then  $z_G^* \leq z_L^* \leq z_H^* \leq z_D^* \leq 2z_G^*$  (Gribkovskaia et al. 2007).

The *Single Vehicle Hamiltonian 1-M-1-PDP with Combined Demands* is usually called the *Traveling Salesman Problem with Simultaneous Pickups and Deliveries* (TSPSPD). It corresponds to the case [1-M-1|PD|1]. This problem can be reduced to the *Single Vehicle 1-M-1-PDP with Single Demands and Mixed Solutions* through the following transformation. Set the vehicle capacity to  $Q' = Q - \sum_{i \in V} d_{i2}$  and, for each vertex  $i$ , define a pickup amount equal to  $\delta_i = d_{i1} - d_{i2}$  if  $\delta_i \geq 0$ , or a delivery amount  $-\delta_i$  if  $\delta_i \leq 0$ . After this conversion each vertex has a zero delivery or a zero pickup. Note also that the *Single Vehicle 1-M-1-PDP with Single Demands and*



**Fig. 2** An example illustrating four solution shapes for the 1-M-1-PDP with combined demands. (P = pickup, D = delivery, PD = combined pickup and delivery)



*Mixed Solutions* can be solved by any algorithm for the TSPSPD. We thus consider the *Single Vehicle Hamiltonian 1-M-1-PDP with Combined Demands* and the *Single Vehicle 1-M-1-PDP with Single Demands and Mixed Loads* as equivalent and survey the literature on these problems in Sect. 3.2.2.

In the multi-vehicle case, the previous reduction of the *Hamiltonian 1-M-1-PDP with Combined Demands* to the *1-M-1-PDP with Single Demands and Mixed Solutions* fails, since it is not possible to set vehicle capacities in the latter problem. Nevertheless, the *Multi-Vehicle 1-M-1-PDP with Single Demands and Mixed Solutions* can still be seen as a special case of the *Multi-Vehicle Hamiltonian 1-M-1-PDP with Combined Demands*. This problem is reviewed in Sect. 3.1.2

The problem of finding an optimal double-path solution to the *1-M-1 PDP with Combined Demands* reduces to the *1-M-1 PDP with Single Demands and Backhauls* by splitting each vertex of  $V \setminus \{0\}$  into a delivery and a pickup vertex. Algorithms for the double-path *1-M-1 PDP with Combined Demands* for the single vehicle and the multi-vehicle cases are reviewed in Sects. 3.2.1 and 3.2.3, respectively. The literature on general and lasso solutions *1-M-1-PDP with Combined Demands* is surveyed in the following section.

### 3.1.1 General and lasso solutions of the 1-M-1-PDP with combined demands [1-M-1|P-D|–]

Gribkovskaia et al. (2007) describe a number of heuristics yielding general solutions for the *Single Vehicle 1-M-1-PDP with Combined Demands*. These include several construction and improvement procedures as well as a tabu search heuristic. The construction procedures consist of the following. First, a possibly infeasible Hamiltonian solution is obtained by means of a nearest neighbor procedure, a sweep procedure or a modified sweep procedure. Then, using a Hamiltonian tour, a series of general solutions is created by removing one of the edges from it and relinking the vertices in a particular way. Third, a merging procedure attempts to eliminate the second visit to customers that are visited twice by transforming the first visit into a pickup and delivery visit if this is feasible. Two variants of this procedure are possible depending on the direction of merging. Six different constructive procedures emerge from different combinations of these heuristics. An improvement procedure attempts to obtain a better solution by iteratively removing a vertex visited once and reinserting it in the tour. The tabu search heuristic is based on the *Unified Tabu Search Algorithm* of Cordeau et al. (2001) for the VRP. Computational results obtained on 17 instances ranging from 16 to 101 vertices reveal that 62% of solutions are Hamiltonian, 32% contain one customer visited twice, and 6% contain two customers visited twice.

Nowak (2005) has introduced the *Pickup and Delivery Problem with Split Loads* (PDPSL), a problem very similar to the general 1-M-1-PDP with combined demands. The only difference is that delivery and pickup requests are allowed to be split into any number of visits. This problem is an extension of the *Split Delivery VRP* introduced by Dror and Trudeau (1989). Nowak concludes that routing costs can be reduced by up to one half by allowing split loads. More recently, Katoh and Yano (2006) have developed a polynomial time approximation algorithm with a worst-case performance ratio of 2 for the PDPSL in which each vehicle has a unit capacity and the network is a tree.

Gribkovskaia et al. (2001) have developed heuristics to generate lasso solutions for the single vehicle and multi-vehicle *1-M-1-PDP with Combined Demands*. Hoff and Løkketangen (2006) have developed a tabu search heuristic to generate lasso solutions in the *Single Vehicle 1-M-1-PDP with Combined Demands*. The neighborhood is defined as the set of solutions which can be reached with the 2-opt operator. The authors present results on instances with up to 262 vertices. Hoff et al. (2006) have later developed another tabu search algorithm for the lasso 1-M-1-PDP but for the multi-vehicle case. The search algorithm defines as valid moves a restricted set of two classes of actions. The first class consists of moving a vertex from one route to another, while the second class consists of swapping two vertices from different routes. A load parameter in the algorithm allows to control the path length and can be used to solve the *Hamiltonian 1-M-1-PDP with Combined Demands* as well as the *1-M-1-PDP with Single Demands and Backhauls*. The algorithm was tested on instances with up to 262 vertices and 25 vehicles.

### 3.1.2 The multi-vehicle Hamiltonian 1-M-1-PDP with combined demands [1-M-1|PD|m]

The *Multi-Vehicle Hamiltonian 1-M-1-PDP with Combined Demands* is usually called the *Vehicle Routing Problem with Simultaneous Pickup and Delivery* (VRPSPD). One result of primary theoretical importance regarding the VRPSPD is due to Anily (1996) who proposed an  $O(n^3 \log n)$  heuristic algorithm for the VRPSPD that converges to the optimal solution under some probabilistic conditions. An early three-phase heuristic for the VRPSPD was developed by Min (1989). In the first phase clusters of customers are formed using the average linkage method (Anderberg 1973). The assignment to clusters is performed in the second phase. In the third phase a routing problem is solved by means of a TSP algorithm, iteratively modifying the distance matrix to avoid solutions that exceed vehicle capacities. Another heuristic was developed by Nagy and Salhi (2005) for the VRPSPD with a maximum route length. It consists of constructing an initial solution, which is often infeasible, and then applying a series of improvement and feasibility procedures. The heuristics also work on the multi-depot version of the problem. These heuristics were capable of solving instances of up to 249 customers within a few seconds. Chen and Wu (2006) have proposed an insertion-based procedure and a hybrid heuristic for the VRPSPD. The hybrid heuristic is based on the record-to-record travel principle (Dueck 1993), tabu lists, and several improvement procedures. The algorithms were tested on instances with up to 199 customers and 19 vehicles. Bianchessi and Righini (2007) have developed and compared several constructive, local search and tabu search algorithms for the VRPSPD using five neighborhoods. Several experiments have allowed the authors to conclude that local search with complex and variable neighborhoods are robust and can yield good solutions. A tabu search algorithm for the VRPSPD was presented by Montané and Galvão (2006). The neighborhood is defined by three inter-route movements and a 2-opt intra-route operator. At each iteration, the best feasible neighbor is selected. The meta-heuristic includes intensification and diversification strategies through the use of a frequency penalty scheme. The algorithm yielded good quality solutions on test instances with up to 400 clients within a maximum computing time of six minutes.

Other heuristics were developed by Halse (1992) and Dethloff (2001). The algorithm of Halse (1992) uses a cluster-first- route-second approach. The clustering phase solves an assignment problem, while the routing phase applies a 3-opt local search followed by an exchange improvement procedure. The heuristic of Dethloff (2001) is an extension of the cheapest insertion procedure which takes into account travel distance, residual capacity and radial surcharge in order to avoid the late insertion of remotely located customers. The heuristic was tested on real-life and randomly generated instances.

Dell'Amico et al. (2006) have developed an exact branch-and-price algorithm for the VRPSPD. In order to reduce the computing time of the pricing problem, the authors have designed an upper bounding heuristic and a lower bounding procedure based on state space relaxation. Their algorithm was able to optimally solve instances of up to 40 customers and 22 vehicles within one hour of computing time. An exact algorithm for the VRPSPD with time windows was presented by Angelelli and Mansini (2002). The authors have proposed a branch-and-price procedure based on a set covering formulation for the master problem. Results with the use of different pricing and branching strategies on instances with 20 customers and up to five vehicles are presented.

### 3.2 The 1-M-1-PDP with single demands [1-M-1|P/D|–]

We review the literature of the *1-M-1-PDP with Single Demands* in four sections. The classification criteria are the number of vehicles (single or multiple) and the solution structure which is either mixed or backhaul. As already mentioned at the beginning of Sect. 3, in the *1-M-1-PDP with Single Demands and Mixed Solutions* delivery and pickup customers can be served in any order, while in the *1-M-1-PDP with Single Demands and Backhauls* pickup customers can only be served after all delivery operations have taken place. Section 3.2.1 surveys the *Single Vehicle 1-M-1-PDP with Single Demands and Backhauls*. Section 3.2.3 is devoted to the *Multi-Vehicle 1-M-1-PDP with Single Demands and Backhauls*. Section 3.2.2 deals with the *Single Vehicle 1-M-1-PDP with Single Demands and Mixed Solutions*. Finally, Sect. 3.2.4 describes algorithms for the *Multiple Vehicle 1-M-1-PDP with Single Demands and Mixed Solutions*.

#### 3.2.1 The single vehicle 1-M-1-PDP with single demands and backhauls [1-M-1|P/D|1]

The *Single Vehicle 1-M-1-PDP with Single Demands and Backhauls* is usually called the *Traveling Salesman Problem with Backhauls* (TSPB), a special case of the *Clustered Traveling Salesman Problem* in which  $V \setminus \{0\}$  is partitioned into  $p$  clusters instead of two (Chisman 1975). Note that in this problem the capacity constraint does not play any role, except that the capacity  $Q$  of the vehicle must satisfy  $Q \geq \sum_{i=1}^n d_{i1}$  and  $Q \geq \sum_{i=1}^n (-d_{i2})$  for a solution to exist. The TSPB can be transformed into a TSP by assigning an arbitrarily large constant  $M$  to the cost of all edges connecting delivery and pickup customers. A TSPB heuristic having a worst-case performance ratio of 1.5 is presented by Gendreau et al. (1997). It extends the heuristic proposed by Christofides (1976) for the TSP. It first constructs a minimum spanning tree  $S$

containing one spanning tree for the pickup customers, another for the delivery customers, and two minimum cost edges, one from vertex 0 to the pickup customers and the other from vertex 0 to the delivery customers. It then computes a minimum weight matching  $M$  between the odd degree vertices and constructs a Eulerian graph using the edges of  $S$  and  $M$ . A shortcut technique is applied to obtain a feasible Hamiltonian tour for the TSPB. Gendreau et al. (1996) have developed six heuristic procedures for the TSPB, based on GENIUS (Gendreau et al. 1992). These were tested on several instances ranging from 100 to 1000 vertices. On instances involving up to 200 vertices, all heuristics produced solutions within 8% of the optimal value. In 1997, Mladenović and Hansen (1997) have introduced the variable neighborhood search (VNS) metaheuristic consisting of systematically changing the neighborhood within a local search procedure. These authors have improved the results of Gendreau, Hertz and Laporte for the TSPB by embedding GENIUS within a VNS framework. Ghaziri and Osman (2003) have developed a neural network algorithm for the TSPB based on Kohonen's self-organizing feature maps (Kohonen 1995). The authors have designed a network architecture which evolves into a feasible TSPB tour. The solution obtained is then improved by means of a 2-opt procedure. The accuracy is comparable to those of Gendreau, Hertz and Laporte and of Mladenović and Hansen. Although this algorithm requires more time, the difference in running time between the two algorithms decreases as the problem size increases. More recently, the authors have extended the algorithm to the multi-vehicle case (Ghaziri and Osman 2006).

### 3.2.2 The single vehicle 1-M-1-PDP with single demands and mixed solutions [1-M-1|P/D|1]

The *Single Vehicle 1-M-1-PDP with Single Demands and Mixed Solutions* is usually called the *Traveling Salesman Problem with Pickups and Deliveries* (TSPPD) (Mosheiov 1994). The TSPPD is NP-hard and is strongly related to the TSP. Consider any tour  $(i_1, \dots, i_n)$  on  $V \setminus \{0\}$ . An interesting result proved by Mosheiov (1994) is that there always exists a vertex  $i_l$  such that the tour  $(i_l, i_{l+1}, \dots, i_n, i_1, \dots, i_{l-1}, i_l)$  satisfies the capacity constraint. Thus, a feasible solution for the TSPPD can be constructed by deleting arc  $(i_{l-1}, i_l)$  and connecting the ends of the deleted arc to the depot. Mosheiov (1994) also showed that using a TSP approximation algorithm with an  $\alpha$ -worst-case performance ratio, the resulting TSPPD solution has a worst-case performance of  $(1 + \alpha)$  and the computational complexity is identical to that of the TSP algorithm. The algorithm is called  $PD\alpha T$ . For instance, using the heuristic of Christofides (1976) for the TSP with a 1.5 worst-case ratio bound,  $PD\alpha T$  has a worst-case performance ratio of 2.5 and runs in  $O(n^3)$  time. The author also showed that if locations are uniformly distributed in the Euclidean plane, the heuristic is asymptotically optimal when Karp's (1977) heuristic for the TSP is used.

Mosheiov (1994) has presented a second heuristic called cheapest feasible extension, which is based on an extension of the well-known cheapest insertion heuristic for the TSP (Golden et al. 1980). It first constructs a TSP tour over all delivery customers and the depot using the cheapest insertion heuristic, and then iteratively inserts pickup customers at the best insertion point while maintaining feasibility. Although this algorithm has an unbounded worst-case performance ratio, it often produces better solutions than those obtained with the  $PD\alpha T$  heuristic.

Anily and Mosheiov (1994) have studied the TSPPD under the name *Traveling Salesman Problem with Delivery and Backhauls* and have presented a new bounded heuristic based on the computation of a minimum spanning tree (Kruskal 1956). This heuristic, called 2MST, first constructs a minimum spanning tree in the complete graph and then solves the TSPPD in the tree with an exact and linear algorithm described as follows. The tree is traversed using a depth-first method, visiting first the subtrees with pickup vertices. Delivery vertices are served the first time they are visited, while the others are served after all vertices in the subtree rooted at them have been visited. A Hamiltonian tour is then obtained by following the tree in a depth-first fashion and applying shortcuts. The authors have proved that this solution is feasible with respect to the capacity constraint. The 2MST has a worst-case performance ratio of 2, which is the lowest worst-case performance ratio of any known polynomial time algorithm for the TSPPD.

Gendreau et al. (1999) have proposed two heuristics for the TSPPD. The first is a linear and exact algorithm for the special case where the graph is a cycle. The authors then present a heuristic for the general case based on this exact algorithm. The heuristic, called H, consists of determining a TSP solution using the Christofides heuristic (Christofides 1976), applying the exact TSPPD algorithm on the TSP solution, and then deriving a Hamiltonian solution by using shortcuts. Since this TSPPD path is at most twice as long as the length of the TSP solution, and the Christofides heuristic has a worst-case performance ratio of 1.5, H has a worst-case performance ratio of 3. The authors also prove the tightness of this bound. Although this worst-case performance ratio of 3 is worse than those of the 2MST and the PD $\alpha$ T heuristics, heuristic H has produced better results in several experiments. The second algorithm proposed by Gendreau et al. (1999) is a tabu search heuristic which performs two-arc exchanges on the solution produced by heuristic H. Special care must be taken while performing these exchanges to avoid generating infeasible solutions. Using adequate data structures, the authors were able to check feasibility in constant time. The heuristic solutions improved the solutions produced by heuristic H, and thus outperformed all previous heuristics for the TSPPD at the expense of longer computing times.

Baldacci et al. (2003) have later developed an exact branch-and-cut algorithm for the TSPPD using a new two-commodity flow formulation. The authors were able to optimally solve instances of up to 200 vertices within a limit of one hour of computing time. Süral and Bookbinder (2003) have studied a problem similar to the TSPPD in which the service of customers with pickup requests is optional. Each pickup request gives a certain revenue if it is served, and the objective is to minimize routing costs minus the total revenue obtained from the pickup customers served. The authors have proposed a mixed-integer model based on the Miller–Tucker–Zemlin subtour elimination constraints (Miller et al. 1960) and have considered several tight LP relaxations. Instances with up to 30 vertices were optimally solved. A periodic version of the TSPPD was studied by Alshamrani et al. (2007). In this problem products delivered in one period must then be picked-up in the following period or a penalty cost is incurred. The heuristic consists of first finding a feasible vehicle tour which only takes travel costs into account, and then improving it with the Or-opt operator taking into consideration the penalty costs.

### 3.2.3 The multi-vehicle 1-M-1-PDP with single demands and backhauls [1-M-1|P/D|m]

The *Multi-Vehicle 1-M-1-PDP with Single Demands and Backhauls* is usually called the *Vehicle Routing Problem with Backhauls* (VRPB). The VRPB is NP-Hard since it includes the capacitated *Vehicle Routing Problem* (VRP) as a special case. Several heuristic and exact algorithms have been proposed for the VRPB over the last 20 years.

The first heuristic for the VRPB was probably introduced by Deif and Bodin (1984). It was designed for symmetric instances and is an extension of the Clarke and Wright (1964) algorithm for the VRP. The authors have modified the classical saving definition to account for the presence of linehaul and backhaul vertices in the merged route. The algorithm cannot control the number of routes and may, therefore, return an infeasible solution. Another heuristic for the VRPB with Euclidean distances was proposed by Goetschalckx and Jacobs-Blecha (1989). Their algorithm uses space filling curves, first described by Peano (1890) and used in a heuristic for solving the planar TSP (Bartholdi and Platzman 1982). The authors have used a space filling curve for transforming pickup and delivery locations in the plane into points on a line. Using a greedy procedure for solving a series of  $m$ -median problems, the pickup and delivery points on the line are partitioned separately forming clusters of routes. Finally, each route of delivery customers is merged with the nearest backhaul route. Results show that the heuristic is generally faster than that of Deif and Bodin, but the solutions are often worse. Another cluster-first-route-second heuristic for the symmetric and asymmetric VRPB, called TV, was proposed by Toth and Vigo (1999). In this heuristic clusters of linehaul and backhaul customers are first determined by solving a relaxation of the problem which is strengthened in a Lagrangian fashion (for relaxations of the VRPB see Toth and Vigo 2002). A matching between linehaul and backhaul clusters is then computed. Routes are obtained with a modified farthest-insertion heuristic and are improved with intra-route and inter-route post-optimization procedures. The heuristic is relatively fast and almost always obtains better solutions than previous heuristics. Osman and Wassan (2002) have developed a tabu search heuristic obtaining solutions with similar quality but requiring more computing time. Brandão (2006) has proposed a much faster tabu search algorithm which uses a VRPB relaxation called  $m$ -tree as an initial solution. The algorithm has yielded the best known results for the VRPB within a relatively small amount of computing time.

Exact algorithms for the VRPB have been presented by Yano et al. (1987), Toth and Vigo (1997), and Mingozzi et al. (1999). Motivated by a problem in a chain of retail stores, Yano et al. (1987) have proposed an exact algorithm for a special case of the VRPB in which the number of pickup and delivery customers in a route does not exceed four. The authors have modeled this particular VRPB as a set covering problem and developed a branch-and-bound algorithm. Toth and Vigo (1997) have presented a branch-and-bound procedure for solving the symmetric and the asymmetric VRPB. In their paper a new Lagrangian lower bound is presented and is straightened by adding valid cuts and combined, in an additive approach (Fischetti and Toth 1989), with another lower bound that results from relaxing the capacity constraints. The lower bound is embedded within a branch-and-bound algorithm. Mingozzi et al.



(1999) have proposed another ILP formulation for the VRPB based on a set partitioned model. They have presented an efficient way of computing a lower bound by combining different heuristic procedures for solving the dual of the LP relaxation of the original model. Their exact method for the VRPB consists of reducing the number of variables of the integer program so that it can be solved with CPLEX. Both algorithms were capable of proving optimality on instances with up to 100 customers and 12 vehicles.

Gélinas et al. (1995) have proposed an exact algorithm for the *VRPPD with Time Windows* (VRPPDTW) which they have extended to the *VRPB with Time Windows* (VRPBTW). Using a set partitioning model, the authors have developed a branch-and-bound algorithm based on column generation. Instead of branching on flow variables, they branch on resource variables such as time or capacity. This branching scheme was proved to be superior on a series of instances with up to 100 customers. Heuristic algorithms for the VRPBTW have been described in three papers. Potvin et al. (1996) have developed a genetic procedure designed to find a good ordering of customers to be used by a route construction heuristic. A tabu search algorithm was developed by Duhamel et al. (1997), in which the neighborhood is randomly chosen between 2-opt, Or-opt, and a swapping procedure. Thangiah et al. (1996) have proposed a constructive heuristic which inserts customers into routes one by one, followed by either of two local search procedures. The first, called  $\lambda$ -interchange, is based on the interchange of customers between routes. The second is a 2-opt\* procedure which is a VRPTW extension of the 2-opt algorithm. The three algorithms have yielded near optimal results on instances of up to 100 customers and 21 vehicles. Ropke and Pisinger (2006b) have introduced a problem called the *Rich Pickup and Delivery Problem with Time Windows* and have developed a large neighborhood search (LNS) heuristic for it. The authors have shown how their heuristic can be used to solve a collection of 1-M-1-PDPs including the *1-M-1-PDP with Single Demands and Backhauls*, and the *1-M-1-PDP with Single Demands and Mixed Solutions*, with or without time windows. The heuristic uses six vertex removal procedures and five insertion procedures. A learning layer modifies the probability distribution of selecting a particular procedure at each iteration according to its past performance.

### 3.2.4 The multi-vehicle 1-M-1-PDP with single demands and mixed solutions [1-M-1|P/D|m]

As explained at the beginning of Sect. 3, the *Multi-Vehicle 1-M-1-PDP with Single Demands and Mixed Solutions* is a special case of the *Multi-Vehicle Hamiltonian 1-M-1-PDP with Combined Demands*. Therefore, the algorithms surveyed in Sect. 3.1.2 can be also applied to the problem of this section. We will now describe some of the most important algorithms designed for the *Multi-Vehicle 1-M-1-PDP with Single Demands and Mixed Solutions*.

One of the first heuristics for the *Multi-Vehicle 1-M-1-PDP with Single Demands and Mixed Solutions*, usually called the Mixed VRPB, was developed by Golden et al. (1988). It is based on the insertion of backhaul customers into routes formed by linehaul customers. The authors have introduced a penalty in the insertion cost function which takes into account the number of linehaul customers on the route segment after the insertion point. A similar approach, called “load-based insertion procedure” was

proposed by Casco et al. (1988). In this algorithm the penalty in the insertion cost accounts for the number of commodities that must be delivered after the insertion rather than the number of remaining delivery customers. Salhi and Nagy (1999) have extended the idea of Casco, Golden and Wasil by introducing a less shortsighted insertion heuristic in which clusters of backhaul customers are inserted together. The algorithm, which yields small improvements, was also adapted in a simple way to solve problems with multiple depots.

Heuristics for the mixed VRPB with time-windows have also been put forward. Kontoravdis and Bard (1995) have developed a GRASP for the mixed VRPB with time windows in which the main objective is to minimize the number of vehicles. At each iteration of the GRASP, a route construction phase is followed by a local search algorithm in which routes can be eliminated. Another two-phase heuristic was proposed by Zhong and Cole (2005). Phase 1 consists of a modified sweep procedure for clustering customers and three route improvement routines running within a guided local search framework (Voudouris and Tsang 1999). Phase 2 iteratively inserts a user-specified number of new routes within an improvement routine in order to achieve a feasible solution. This heuristic provides solutions with a 10% to 19% lower cost compared to those obtained by Kontoravdis, but it generally requires a larger number of vehicles.

Wade and Salhi (2002) have described a heuristic for the *Multi-Vehicle 1-M-1-PDP with Single Demands and Mixed Solutions* in which an additional restriction is imposed. The problem is called RVRPB and the restriction consists of allowing pickup operations only when the percentage of the volume already delivered exceeds a user-controlled parameter  $\alpha$ . If  $\alpha = 100\%$  then there are no restrictions and any solution is permitted. At the other extreme, if  $\alpha = 0\%$ , only backhaul solutions are allowed. Wade and Salhi (2002) have developed a constructive heuristic for obtaining  $\alpha$ -restricted solutions. Their procedure called R-INS first solves the VRP for the delivery customers using the VRP heuristic of Salhi and Rand (1987) and applies a greedy algorithm to insert the pickup customers.

## 4 One-to-one problems

In one-to-one problems each commodity has exactly one pickup vertex and one delivery vertex. These problems are always  $n$ -commodity problems since there exists a one-to-one correspondence between pickup and delivery vertices. Two important problems belonging to this category are the *Vehicle Routing Problem with Pickups and Deliveries* (VRPPD) and the *Dial-a-Ride Problem* (DARP). The VRPPD deals with the transportation of objects, while the DARP applies to the transportation of people. In the latter problem, the objective function usually takes user inconvenience into account. When vehicles are allowed to drop objects temporarily, the problem is called the VRPPD with transshipments (VRPPDT). Another well studied one-to-one problem is the *Stacker Crane Problem* (SCP) (Frederickson et al. 1978). The SCP is a special case of the *Swapping Problem* and of the VRPPD. Sections 4.1, 4.2, 4.3 and 4.4 survey the literature on the SCP, the VRPPD, the DARP and the VRPPDT, respectively.

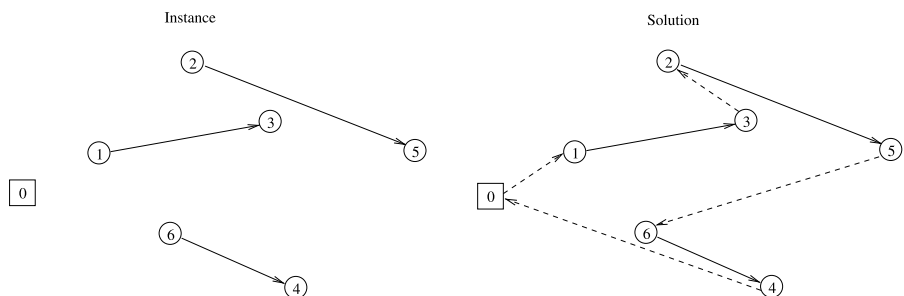


#### 4.1 The stacker crane problem [1-1|P/D|1]

In the SCP single objects have to be transported from their origin to their destination using a unit capacity vehicle. It takes its name from the practical problem of managing crane operations. The SCP contains the following restrictions: (i)  $m = 1$ ; (ii)  $Q = 1$ ; (iii)  $d_{0h} = 0$  for every  $h \in H$ ; (iv) for each vertex  $i \in V \setminus \{0\}$ , there exists exactly one commodity  $h \in H$  such that  $d_{ih} \neq 0$ , and is equal to  $-1$  or  $1$ ; (v) for each commodity  $h \in H$ , there are two vertices  $i, j \in V \setminus \{0\}$  such that  $d_{ih} = 1, d_{jh} = -1$  and  $d_{vh} = 0$  for all  $v \in V \setminus \{i, j\}$ . Restrictions (i) and (ii) mean that the SCP is a unit capacity single vehicle problem. Restriction (iii) indicates that the depot has no supply and no demand. Restrictions (iv) and (v) state that each request, consisting of a single unit supply vertex and a single unit demand vertex, is associated with a different commodity. The SCP is a special case of the *Swapping Problem* (Sect. 2.1). It can be preemptive, non-preemptive or mixed according to the vertex set  $T$  which defines the transshipment points (see Sect. 1). Figure 3 depicts an SCP solution. The SCP, like the SP, is NP-hard (Frederickson and Guan 1992). However, polynomial algorithms are available for certain types of graphs for which the SP still remains NP-hard. A summary of the complexity of the SP and SCP under different types of graphs can be found in Anily et al. (2006).

Frederickson et al. (1978) have developed a heuristic with a worst-case performance ratio of  $9/5$  for the non-preemptive SCP on general graphs. The preemptive SCP on a tree was later studied by Frederickson and Guan (1992) who showed that this problem is polynomially solvable. Specifically, these authors have presented two algorithms running in  $O(k + qn)$  and  $O(k + n \log n)$  time, where  $k$  is the number of objects to be transported,  $n$  is the number of vertices, and  $q \leq \min \{k, n\}$ . Frederickson and Guan (1993) have later studied the non-preemptive SCP on a tree. They have shown that the problem is NP-hard by proving that the *Steiner Tree Problem* can be reduced to it. They have also presented several approximation algorithms with a worst-case performance ratio ranging from 1.5 to around 1.21. The 1.5 worst-case algorithm, based on a Steiner tree approximation, runs in linear time.

Attalah and Kosaraju (1988) have studied the preemptive and non-preemptive SCP on a line and on a circle. Their work was motivated by a robotics problem which consisted of rearranging objects among stations using a robot arm. The arm is a telescope link that rotates around a fixed point. Since minimizing the total distance traveled by



**Fig. 3** Example of a non-preemptive SCP instance and a solution

the robot arm is NP-hard, the authors have studied the problem of minimizing separately the angular and the telescoping movements. The minimization of the angular motion is equivalent to the minimization problem on a circle, while the minimization of the telescoping motion corresponds to the SCP on a line. For the non-preemptive case, the authors have developed an  $O(k + n \log n)$  algorithm for the circular problem and an  $O(k + n\alpha(n))$  algorithm for the line problem, where  $\alpha$  is the inverse of the Ackermann function (Robinson 1948). The algorithm for the preemptive case runs in  $O(k + n)$  time for the circular and linear cases.

#### 4.2 The vehicle routing problem with pickups and deliveries [1-1|P/D|—]

The *Vehicle Routing Problem with Pickups and Deliveries* (VRPPD) consists of routing a fleet of vehicles in order to satisfy a set of customer requests. The requests specify the size of the load to be transported as well as the pickup and delivery locations. Each request needs to be served by one vehicle and the pickup location must be visited before the delivery location. An important example where the VRPPD is applied is found in the routing operations of local area courier services. Another rich application area is encountered in the design of tramp shipping routes (Brønmo et al. 2007). The VRPPD has been studied for more than 30 years and, probably, constitutes the most popular type of PDP. When transportation requests are for people, a maximum ride time constraint is typically imposed, and the problem is called the DARP (see Sect. 4.3). The VRPPD is NP-hard since it generalizes the VRP. The VRPPD contains the following restrictions: (i)  $d_{0h} = 0$  for every  $h \in H$ ; (ii) for each vertex  $i \in V \setminus \{0\}$ , there exists exactly one commodity  $h \in H$  such that  $d_{ih} \neq 0$ ; (iii) for each commodity  $h \in H$ , there are two vertices  $i, j \in V \setminus \{0\}$  such that  $d_{ih} > 0$ ,  $d_{jh} = -d_{ih}$  and  $d_{vh} = 0$  for all  $v \in V \setminus \{i, j\}$ ; (iv)  $T = \emptyset$ . Restriction (i) forces the depot not to demand or supply any commodity. Restrictions (ii) and (iii) state that each request is associated with a different commodity. Finally, restriction (iv) disallows temporary drops.

The VRPPDTW has been well studied. In the VRPPDTW, if a vehicle arrives at a vertex before the beginning of its time window it must wait until that time to begin its service. Deciding whether an instance of the VRPPDTW is feasible is NP-complete (Savelsbergh 1985). The VRPPD with soft time windows gives rise to a scheduling problem (Sexton and Bodin 1985a; Dumas et al. 1990). Scheduling in the VRPPD and VRPPDTW is also important in the dynamic versions of the problems (Mitrović-Minić and Laporte 2004; Branke et al. 2005).

##### 4.2.1 The single vehicle routing problem with pickups and deliveries [1-1|P/D|1]

The *Single Vehicle Routing Problem with Pickups and Deliveries* (SVRPPD) is a special case of the VRPPD, in which there is only one vehicle. One of the first studies on the SVRPPD was made by Stein (1978) for the uncapacitated case. Extending the results of Beardwood et al. (1959), the author has proved that if  $n$  pickup and delivery pairs are chosen randomly with a uniform distribution on a square and on an Euclidean region of area  $a$ , the length  $Y_n$  of an optimal tour satisfies

$$\lim_{n \rightarrow \infty} \frac{Y_n}{\sqrt{n}} = \frac{4}{3} \sqrt{2ab}, \quad (1)$$

where  $b \approx 0.713$  is the TSP constant. The author has proposed a simple heuristic consisting of building a TSP tour for the pickup requests, a TSP tour for the delivery requests, and then concatenating the two tours. Under the same hypotheses, the author proved that the heuristic has an asymptotic performance ratio of 1.06. Another heuristic for the uncapacitated SVRPPD was proposed by Psaraftis (1983a). It first constructs a TSP tour through all the pickup and delivery vertices, and then a SVRPPD solution is obtained by traversing the TSP clockwise until all vertices are visited. Vertices already visited or delivery vertices, for which the corresponding pickup vertex is not yet visited, are skipped. The author proved that if the TSP heuristic of Christofides (1976) is used in the first step, the heuristic runs in  $O(n^2)$  time and has a worst-case performance ratio of 3. One of the first local search methods for the SVRPPD was the  $k$ -interchange procedure introduced by Psaraftis (1983b), which extends the idea proposed by Lin (1965) for the TSP. In the SVRPPD pickup-delivery precedence constraints must be checked for each possible interchange. Psaraftis has shown how to find the best SVRPPD  $k$ -interchange in  $O(n^k)$  time, the same order as for the TSP. Another efficient implementation of local search algorithms for the TSP with side constraints is due to Savelsbergh (1990). For an experimental comparison of classical construction heuristics and local search methods for the capacitated SVRPPD, see Kubo and Kasugai (1990). Van der Bruggen et al. (1993) have constructed a heuristic procedure for the capacitated SVRPPD with time windows. Their local search method, which is similar to the Lin and Kernighan (1973) algorithm for the TSP, consists of two phases. First, a feasible solution is obtained by visiting the locations in order of increasing centers of their time window, while taking into account precedence and capacity constraints. If the solution violates time windows, an iterative arc-exchange procedure is applied in hope of obtaining a feasible solution. The algorithm was tested on instances of up to 50 vertices and the solutions obtained were near-optimal. An extension of a 2-opt local search method for the SVRPPD has been proposed by Healy and Moll (1980). This heuristic, called “sacrificing”, consists of escaping from local optima by considering a second metric cost, which is defined by the size of the feasible neighborhood. Once a local optimum is reached, the procedure finds a worse solution with a larger neighborhood and then continues with the standard local search. Experimental results on instances of up to 100 requests indicate that applying sacrificing after a 2-interchange procedure considerably improves the quality of solutions.

Exact algorithms for the SVRPPD were developed by Desrosiers et al. (1986), Kalantari et al. (1985), Fischetti and Toth (1989), and Ruland and Rodin (1997). Desrosiers et al. (1986) have proposed an exact dynamic programming algorithm for the capacitated SVRPPD with time windows. Each state is an ordered pair  $(S, i)$ , corresponding to a feasible route beginning at the depot, visiting all vertices in  $S \subseteq V$ , and finishing at vertex  $i \in S$ . States use a two-dimensional labeling with the time and cost of that partial solution. The algorithm efficiently eliminates infeasible and dominated states. Computational experiments have shown that this algorithm was able to solve instances with up to 40 vertices in a very short time.

Kalantari et al. (1985) have developed branch-and-bound procedures for the capacitated and uncapacitated versions of the SVRPPD. Their procedures, which are extensions of the algorithm for the TSP of Little et al. (1963), eliminate at each node

of the search tree all the arcs that would produce solutions violating some precedence constraints. Computational results solving to optimality instances of up to 15 requests were presented. Fischetti and Toth (1989) have proposed an additive approach to compute lower bounds. The authors have applied their algorithms to the TSP with precedence constraints in which tour solutions must respect a given partial order between nodes. The bounds used for this problem are based on the shortest spanning tree, the assignment problem, variable decomposition, and disjunctions. The exact procedure presented by Ruland and Rodin (1997) for the capacitated SVRPPD consists of a branch-and-cut algorithm using four classes of inequalities as cuts. The method is similar to the TSP algorithm of Padberg and Rinaldi (1991).

The special case of the SVRPPD, called *TSPPD with LIFO Loading* (TSPPDL), arising when the deliveries must respect a last-in-first-out rule of service, was studied by Pacheco (1997), Carrabs et al. (2006), and Cordeau et al. (2006). The LIFO service means that the vehicle can only make a delivery associated with the last pickup operation. This restriction appears, for instance, in the routing of rear-loading vehicles and of automated guided vehicles (AGVs) that use a stack to move items between locations. Pacheco has proposed a heuristic for the uncapacitated TSPPDL, based on the Or-opt algorithm (Or 1976) for the TSP. Carrabs et al. (2006) have introduced three new local search operators for the problem, which they have embedded within a VNS framework. A branch-and-cut algorithm for this problem was developed by Cordeau et al. (2006). The authors have presented three ILP formulations and have improved their linear relaxations using valid inequalities. Instances with up to 25 requests were solved within a reasonable computing time.

Lübbecke (2004) has studied a particular type of solution to the capacitated SVRPPD, where the sequence of pickup and delivery operations along the tour follows a simple structure. This structure is the concatenation of pickup and delivery routes with one or two requests. This special type of solutions is called simple pickup and delivery paths. Assuming  $p_i$  and  $p_j$  are any two pickup locations, and  $d_i$  and  $d_j$  are their corresponding delivery locations, a simple pickup and delivery path is created from the concatenation of the following types of paths:  $(p_i, d_i)$ ,  $(p_i, p_j, d_j, d_i)$ , and  $(p_i, p_j, d_i, d_j)$ . The author proved that finding the optimal simple pickup and delivery path is still NP-hard and presented a special case where the optimal solution can be found in polynomial time.

#### 4.2.2 The multi-vehicle routing problem with pickups and deliveries [1-1|P/D|m]

An exact column generation algorithm for the VRPPD with time windows was proposed by Dumas et al. (1991). The algorithm, which can also handle multiple depots and different types of vehicles, can solve instances with up to 55 requests under tight capacity constraints. Savelsbergh and Sol (1998) have developed another exact algorithm for the VRPPD with time windows, similar to that of Dumas, Desrosiers and Soumis. The algorithm was designed to be embedded in the decision support system of a distribution company and is capable of inserting new requests dynamically as they appear. Xu et al. (2003) have proposed another column generation algorithm for a VRPPD possessing a series of restrictions commonly encountered in real-world logistics. Examples of these are requests with multiple time windows, driver work

rules, compatibility constraints between carriers, vehicles and requests, etc. The master problem is solved by an LP solver, while the subproblems are solved by a heuristic. The algorithm was capable of yielding near-optimal solutions to randomly generated instances with up to 200 requests within a reasonable time. Recently, a branch-and-cut-and-price algorithm for the VRPPDTW was proposed by Ropke and Cordeau (2007). The VRPTW is formulated as a set partitioning problem, and two shortest path problems are considered as pricing subproblems. The authors have shown that the LP relaxation of the set partitioning formulation implies several valid inequalities previously used by Ropke et al. (2007) and Cordeau (2006) within branch-and-cut algorithms. The algorithm was capable of optimally solving some instances with up to 500 requests.

A reactive tabu search heuristic (Battiti and Tecchiolli 1994) for the VRPPDTW was developed by Nanry and Barnes (2000). A solution is first obtained using a greedy procedure. Three types of neighbor moves were considered. The first move consists of moving a pickup-delivery pair from one route to another. The second type of move consists of swapping pickup-delivery pairs between routes. The third move reorders customers within a route. During the search infeasible solutions that do not respect the vehicle capacity or the time windows are allowed. The objective function in the heuristic is modified to include weighted penalty terms. A hierarchical search mechanism was developed in order to change between the three neighborhoods according to how tight time windows constraints are in the current solution. Optimal or near-optimal results were obtained in less than five minutes on a series of random instances of 100 customers.

Lau and Liang (2002) have developed a two-phase heuristic procedure for the VRPPD, in which the number of vehicles is not specified. The objective consists of a weighted combination of the number of vehicles used and routing cost. The first phase combines a construction and a sweep procedure. The second phase applies a tabu search with three different neighborhood moves similar to those of Nanry and Barnes.

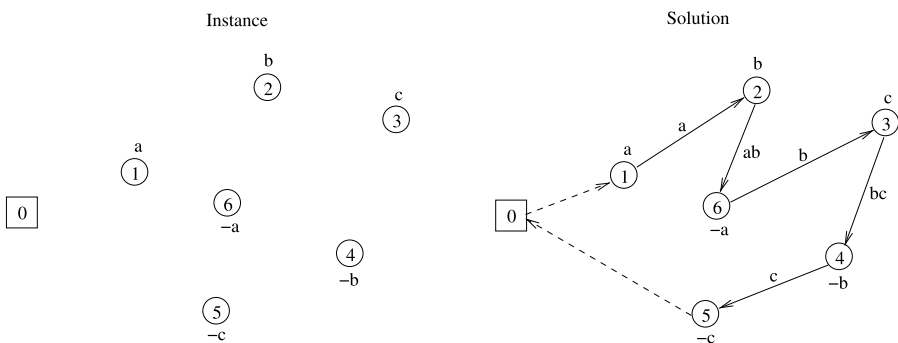
A two-stage heuristic for the VRPPDTW was developed by Bent and Van Hentenryck (2006). Their objective was first to minimize the number of used vehicles and then travel costs. The first stage consists of a simulating annealing algorithm which minimizes the number of routes. A large neighborhood search (LNS) algorithm is then applied in the second stage. The neighborhood used in this LNS is defined as the set of solutions that can be reached by relocating at most  $p$  requests. The heuristic has produced several new best solutions on instances with 100, 200 and 600 customers within at most 90 minutes of computing time. Another LNS heuristic for the VRP-PDTW was later proposed by Ropke and Pisinger (2006a). As in the algorithm of Ropke and Pisinger (2006b) for the 1-M-1-PDP, the selection of removal and insertion procedures is made through a self-adjusting mechanism, which attributes a score to their previous performance. The heuristic has generally produced better solutions than those of Bent and Van Hentenryck (2006) using approximately the same computing time. Two new formulations and a branch-and-cut algorithm for the VRP-PDTW and DARP were developed by Ropke et al. (2007). This article is reviewed in Sect. 4.3.2.

### 4.3 The dial-a-ride problem $[1-1|P/D|-]$

The DARP is a special case of the VRPPD in which requests are made by users needing to be transported from an origin to a destination. The DARP can have several constraints such as time windows, maximum ride times, and other quality of service related restrictions (Cordeau and Laporte 2003a, 2003b; Cordeau 2006). The main application of the DARP is the door-to-door transportation services offered for the elderly and handicapped people in many cities. Case studies have been conducted in Toronto (Desrosiers et al. 1986), Berlin (Borndörfer et al. 1997), Bologna (Toth and Vigo 1996), Copenhagen (Madsen et al. 1995), and Brussels (Rekiek et al. 2006). The minimization of user inconvenience often has to be balanced with operation costs since these objectives usually conflict. User inconvenience is taken into consideration, for instance, by assigning time windows to pickup or deliveries and setting a maximum trip time for each user. Paquette et al. (2006) have conducted a comparative study of several models that give different priorities to system cost and user inconvenience. The dial-a-ride systems can also constitute a good way of providing transportation services at reasonable cost in rural and urban regions with low population density (Mageean and Nelson 2003).

#### 4.3.1 The single vehicle dial-a-ride problem $[1-1|P/D|1]$

One of the first studies of the single vehicle DARP (Fig. 4) was carried out by Psaraftis (1980) who examined the immediate-request case, in which users wish to be serviced as soon as possible. The objective is to minimize a weighted combination of the total service time and user dissatisfaction. The level of dissatisfaction is a linear function of the waiting time for pickup and ride time. The problem was solved exactly by dynamic programming. The algorithm has a complexity of  $O(n^2 3^n)$  and can thus solve only small instances. In a later paper Psaraftis (1983b) extended the algorithm to handle hard time windows at pickup and delivery locations. Sexton and Bodin (1985a, 1985b) have proposed a heuristic algorithm based on Benders decomposition for the single vehicle DARP. In their problem, users specify a desired delivery time and lateness is not allowed. The objective is to minimize total user inconvenience



**Fig. 4** A DARP instance and solution with three requests. Each request consists of an origin vertex denoted by a letter, and a destination vertex denoted by the same letter preceded by a minus sign. Requests being served while the vehicle is traversing an arc are denoted by labels on the arc

which consists of two elements. The first is the delivery time deviation, defined as the difference between the desired delivery time and the actual delivery time. The second measure is the excess ride time, which is the difference between the actual ride time and the shortest ride time. The algorithm decomposes the problem into routing and scheduling subproblems. The authors have shown that the dual of the scheduling subproblem is a network flow problem, and can thus be solved efficiently. The routing problem is solved by a heuristic based on the Benders master problem. The algorithm was tested on real problems with up to 20 users.

#### 4.3.2 *The multi-vehicle dial-a-ride problem* [1-1|P/D|m]

In the multi-vehicle version of the DARP an assignment of requests to vehicles must be performed in addition to the design and scheduling of each route. As a consequence, the solution space is considerably larger than in the single vehicle DARP and, therefore, the problem becomes much more difficult to solve.

Jaw et al. (1986) have developed a heuristic procedure for the multi-vehicle DARP, where users specify either a desired pickup time or a desired delivery time, but not both. Customers must be picked-up or delivered after or before the stated time. Vehicles are not allowed to be idle while carrying users, and the objective consists of finding a compromise between cost minimization and service maximization, where service is measured by the duration of ride times and the deviations from desired pickup and delivery times. The algorithm works by inserting users in order of their earliest feasible pickup time into vehicles routes. At each step the algorithm considers all feasible ways of inserting a user in every route, and the move yielding the minimum additional cost is implemented. The authors have also developed a version of their algorithm in which more than one user is considered at the same time. Computational tests were carried out on artificial and real instances with up to 2617 users and 28 vehicles.

A modification to the heuristic of Jaw et al. (1986) was later proposed by Potvin and Rousseau (1992), who have modified the insertion procedure by keeping the best partial solutions at each iteration. This method reduces the myopic behavior of the previous heuristic. In addition, two new phases were proposed. One is an optional initialization phase that clusters some customers to create partial routes. The other is a post-optimization procedure based on customer exchanges. This heuristic produces better results than that of Jaw et al. but requires larger computational time.

Cullen et al. (1981) have developed an interactive routing heuristic for the DARP and other routing problems without time windows using a cluster-first-route-second technique. This class of algorithms first clusters customers and then designs a route for each cluster. In their heuristic Cullen, Jarvis and Ratliff solve the clustering and the routing subproblems by column generation.

Another heuristic for the multi-vehicle DARP was proposed by Roy et al. (1983) who have studied a complex version of the DARP including the following features. Requests may be to carry one or more people, with or without a wheelchair or an accompanying person, multiple service quality measures may apply, vehicle speed may vary over time, etc. Users specify either a desired pickup time (inbound requests) or a desired delivery time (outbound requests). The objective is first to minimize the total operation costs ensuring a minimum quality of service. In a post-optimization



phase, the algorithm attempts to maximize the quality of service without increasing the operational costs reached in the first phase. This heuristic is also a cluster-first-route-second method. Clusters of users are created according to a proximity relation. An affinity measure is then used to guide the parallel insertion of clusters of requests into routes. This algorithm was successfully tested on real instances with up to 578 requests. Bodin and Sexton (1986) have developed another cluster-first-route-second heuristic for the DARP with desired delivery times, where the objective is to minimize the sum of the total delivery deviation and excess ride time. The heuristic first partitions customers into clusters and constructs a tour on each cluster using the single vehicle DARP algorithm of Sexton and Bodin (1985a, 1985b). Finally a method called “swapper” attempts to move customers between routes and performs route re-optimizations.

Dumas et al. (1989) have modified the cluster-first-route-second methodology with the use of mini-clusters representing a set of users wishing to travel at more or less the same time and within the same subregion. In the formation of mini-clusters only local temporal and spatial considerations are necessary. The authors have proposed a heuristic for the multi-vehicle DARP consisting of the following steps. In the first step, a heuristic procedure extracts a set of mini-clusters from a feasible solution obtained by a simple heuristic, such as the one of Roy et al. (1983). Second, an exact column generation algorithm is used to optimally combine mini-clusters into vehicle routes. Third, each vehicle route is reoptimized using the single vehicle DARP algorithm of Desrosiers et al. (1986). Finally, an optimal schedule that minimizes the total inconvenience costs of users is obtained for each route using the scheduling algorithm of Dumas et al. (1990). Another way of constructing mini-clusters was proposed by Desrosiers et al. (1991) who used a parallel insertion method based on neighboring requests. Their notion of neighboring requests takes into account temporal and spatial proximity. Ioachim et al. (1995) have later applied an optimization technique instead of a heuristic procedure for the creation of mini-clusters. The authors have embedded the mini-cluster generation problem within a set partitioning formulation which was solved by column generation. On a real instance with 2545 requests the new method outperformed the mini-cluster parallel insertion procedure of Desrosiers et al. (1991) by almost 6%.

A sophisticated version of the DARP arising from the case of Bologna was studied by Toth and Vigo (1996). In this version of the DARP, users have different service requirements according to their disabilities, and time windows at pickup and delivery locations are imposed by the users. In addition to a fleet of vehicles, the use of taxis is allowed. The objective is to minimize total operational costs. The authors have presented a fast local search algorithm that considerably improves the solutions obtained with a parallel insertion procedure. The local search uses many neighborhood spaces arising from intra-route and inter-route moves and exchanges. Improvements were observed in computational experiments on real-life instances with about 300 customers. Toth and Vigo (1997) have later replaced their local search algorithm with a tabu thresholding algorithm (Glover 1995).

Cordeau and Laporte (2003b) have developed a tabu search heuristic for the DARP with time windows, where the objective is to minimize routing costs. During the search infeasible solutions are allowed through the use of a relaxation mechanism with self-adjusting penalty coefficients. The search algorithm works by removing



a request from one route and inserting it into another. During a candidate evaluation a minimum duration schedule that does not increase the time window and ride-time violations is computed. The algorithm was tested on randomly generated instances with up to 144 requests and on real-life instances having 200 and 295 requests. A branch-and-cut algorithm was later developed by Cordeau (2006) for the same problem. This algorithm uses several new valid inequalities for the DARP as well as known inequalities for other vehicle routing problems. The algorithm was able to solve to optimality instances of up to 32 requests and four vehicles within a computing time limit of four hours. Another branch-and-cut algorithm for the DARP and the VRPPDTW was developed by Ropke et al. (2007). The authors have adapted known constraints to the new formulations and they have introduced two new families of inequalities. Experimental results have shown that these approaches are considerably faster than the one used in Cordeau (2006).

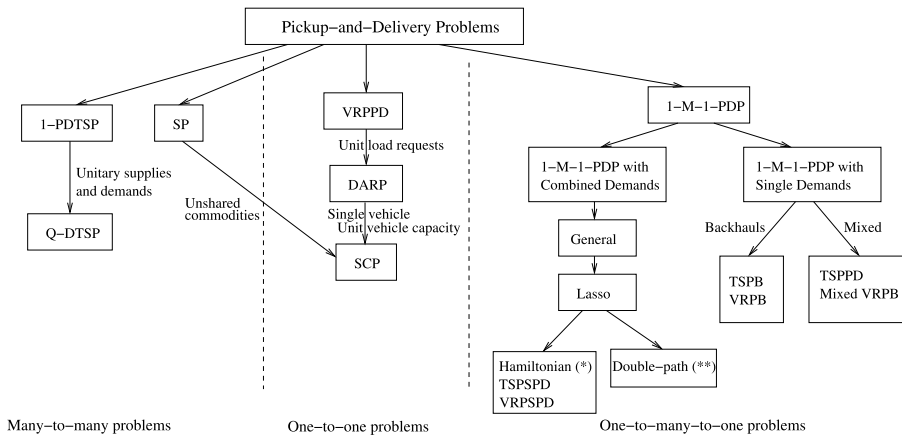
#### 4.4 The vehicle routing problem with pickups, deliveries and transshipments [1-1|P/D-T|-]

In the *Vehicle Routing Problem with Pickups, Deliveries and Transshipments* (VRP-PDT) objects can be transported by different vehicles on their path from the source to the destination. This is achieved by letting vehicles drop and pickup objects at specific points called transshipment (or transfer) points. The VRPPDT contains the same restrictions as the VRPPD except that the set of transfer points is not empty. Due to the existence of transshipment points, the number of solutions of the VRPPDT is much larger than in the standard VRPPD. This flexibility allows the satisfaction of more requests but poses the challenge of creating efficient methods to find optimal solutions in such a large solution space.

Cortés et al. (2007) have proposed a mixed integer linear programming model for the dial-a-ride version of the VRPPDT with time windows. The authors have implemented a branch-and-cut algorithm based on a decomposition method and solved very small instances with up to eight customers and one transshipment point. Finally, Mitrović-Minić and Laporte (2006) have developed a two-phase heuristic algorithm for the VRPPDT with time windows. The first phase uses cheapest insertions, and the second is an improvement phase based on request reinsertions. Requests can be split within both phases to allow transshipment. The heuristic has solved randomly generated instances with up to 100 requests and four transshipment points. The authors have concluded that significant gains can be obtained by allowing transfers when requests are clustered.

## 5 Conclusions

We have provided an overview of static pickup and delivery problems (PDPs). The main relationships between these problems are depicted in Fig. 5. We have proposed a general framework embedding a large collection of PDPs. The literature on these problems is growing fast with the advent and development of new sophisticated exact and heuristic algorithms. We hope this survey will stimulate further research in this area.



**Fig. 5** A classification scheme for pickup-and-delivery problems. \* The single vehicle *Hamiltonian 1-M-1-PDP with combined demands* is equivalent to the single vehicle *1-M-1-PDP with single demands and mixed solutions*. \*\* The *1-M-1-PDP with combined demands and double-path solution* is equivalent to the *1-M-1-PDP with single demands and backhauls*

**Acknowledgements** This work was supported by the ministère de l'Éducation, du Loisir et du Sport du Québec, and by the Canadian Natural Sciences and Engineering Research Council under grants 227837-04 and 39682-05. This support is gratefully acknowledged. Thanks are due to five discussants and to Krystina Bakhrankova for their valuable comments.

## References

- Alshamrani A, Mathur K, Ballou RH (2007) Reverse logistics: simultaneous design of delivery routes and return strategies. *Comput Oper Res* 34:595–619
- Anderberg MR (1973) Cluster analysis for applications. Academic, New York
- Angeles E, Mansini R (2002) The vehicle routing problem with time windows and simultaneous pick-up and delivery. In: Speranza MG, Klose A, Van Wassenhove LN (eds) *Quantitative approaches to distribution logistics and supply chain management*. Springer, Berlin, pp 249–267
- Anily S (1996) The vehicle routing problem with delivery and back-haul options. *Nav Res Logist* 43:415–434
- Anily S, Bramel J (1999) Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries. *Nav Res Logist* 46:654–670
- Anily S, Hassin R (1992) The swapping problem. *Networks* 22:419–433
- Anily S, Mosheiov G (1994) The traveling salesman problem with delivery and backhauls. *Oper Res Lett* 16:11–18
- Anily S, Gendreau M, Laporte G (1999) The swapping problem on a line. *SIAM J Comput* 29:327–335
- Anily S, Gendreau M, Laporte G (2006) The preemptive swapping problem on a tree (submitted for publication)
- Attalah M, Kosaraju S (1988) Efficient solutions to some transportation problems with applications to minimizing robot arm travel. *SIAM J Comput* 17:849–869
- Balas E, Padberg MW (1976) Set partitioning: a survey. *SIAM Rev* 18:710–760
- Baldacci R, Hadjiconstantinou EA, Mingozzi A (2003) An exact algorithm for the traveling salesman problem with deliveries and collections. *Networks* 42:26–41
- Bartholdi JJ, Platzman LK (1982) An  $O(n \log n)$  planar travelling salesman heuristic based on space filling curves. *Oper Res Lett* 4:121–125
- Battiti R, Tecchioli G (1994) The reactive tabu search. *ORSA J Comput* 6:126–140

- Beardwood J, Halton JH, Hammersley JM (1959) The shortest path through many points. *Proc Camb Philos Soc* 55:299–328
- Bent R, Van Hentenryck P (2006) A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Comput Oper Res* 33:875–893
- Beullens P, Van Wassenhove L, Van Oudheusden D (2004) Collection and vehicle routing issues in reverse logistics. In: Dekker R, Fleischmann M, Inderfurth K, Van Wassenhove L (eds) *Reverse logistics, quantitative models for closed-loop supply chains*. Springer, Berlin, pp 95–134
- Bianchessi N, Righini G (2007) Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Comput Oper Res* 34:578–594
- Bodin LD, Sexton TR (1986) The multi-vehicle subscriber dial-a-ride problem. *TIMS Stud Manag Sci* 22:73–86
- Borndörfer R, Klostermeier F, Grötschel M, Küttner C (1997) *Telebus Berlin: vehicle scheduling in a dial-a-ride system*. Technical Report SC 97-23, Konrad-Zuse-Zentrum für Informationstechnik, Berlin
- Brandão J (2006) A new tabu search algorithm for the vehicle routing problem with backhauls. *Eur J Oper Res* 173:540–555
- Branke J, Middendorf M, Noeth G, Dessouky M (2005) Waiting strategies for dynamic vehicle routing. *Transport Sci* 39:298–312
- Brønmo G, Christiansen M, Fagerholt K, Nygreen B (2007) A multistart local search heuristic for ship scheduling—a computational study. *Comput Oper Res* 34:900–917
- Carrabs F, Cordeau J-F, Laporte G (2006) Variable neighbourhood search for the pickup and delivery traveling salesman problem with LIFO loading. *INFORMS J Comput* (forthcoming)
- Casco DO, Golden BL, Wasil EA (1988) Vehicle routing with backhauls: models, algorithms, and case studies. In: Golden BL, Assad AA (eds) *Vehicle routing: methods and studies*. North-Holland, Amsterdam, pp 127–147
- Chalasani P, Motwani R (1999) Approximated capacitated routing and delivery problems. *SIAM J Comput* 28:2133–2149
- Chen J-F, Wu T-H (2006) Vehicle routing problem with simultaneous deliveries and pickups. *J Oper Res Soc* 57:579–587
- Chisman JA (1975) The clustered traveling salesman problem. *Comput Oper Res* 2:115–118
- Christofides N (1976) Worst-case analysis of a new heuristic for the travelling salesman problem. Research report 388, GSIA, Carnegie Mellon University, Pittsburgh, PA
- Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Oper Res* 12:568–581
- Cordeau J-F (2006) A branch-and-cut algorithm for the dial-a-ride problem. *Oper Res* 54:573–586
- Cordeau J-F, Laporte G (2003a) The dial-a-ride problem (DARP): variants, modeling issues and algorithms. *4OR—Q J Belg Fr Ital Oper Res Soc* 1:89–101
- Cordeau J-F, Laporte G (2003b) A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transport Res Part B* 37:579–594
- Cordeau J-F, Laporte G, Mercier A (2001) A unified tabu search heuristic for vehicle routing problems with time windows. *J Oper Res Soc* 52:928–936
- Cordeau J-F, Iori M, Laporte G, Salazar-González JJ (2006) A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with LIFO loading (submitted for publication)
- Cortés CE, Matamala M, Contardo C (2007) The pickup-and-delivery problem with transfers: formulation and a branch-and-cut solution method (submitted for publication)
- Cullen FH, Jarvis J, Ratliff HD (1981) Set partitioning based heuristic for interactive routing. *Networks* 11:125–143
- Deif I, Bodin L (1984) Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauling. In: Kidder A (ed) *Proceedings of the Babson conference on software uses in transportation and logistic management*, Babson Park, FL
- Dell’Amico M, Righini G, Salani M (2006) A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transport Sci* 40:235–247
- Desrosiers J, Dumas Y, Soumis F (1986) A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *Am J Math Manag Sci* 6:301–325
- Desrosiers J, Dumas Y, Soumis F, Taillefer S, Villeneuve D (1991) An algorithm for mini-clustering in handicapped transport. Technical Report G-91-02. GERAD, HEC Montréal
- Dethloff J (2001) Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *Oper Res Spektrum* 23:79–96
- Dror M, Trudeau P (1989) Savings by split delivery routing. *Transport Sci* 23:141–145
- Dueck G (1993) New optimization heuristics: the great deluge algorithm and the record-to-record travel. *J Comput Phys* 104:86–92

- Duhamel C, Potvin J-Y, Rousseau J-M (1997) A tabu search heuristic for the vehicle routing problem with backhauls and time windows. *Transport Sci* 31:49–59
- Dumas Y, Desrosiers J, Soumis F (1989) Large scale multi-vehicle dial-a-ride problems. Research report G-89-30, GERAD, HEC Montréal
- Dumas Y, Soumis F, Desrosiers J (1990) Optimizing the schedule for a fixed vehicle path with convex inconvenience costs. *Transport Sci* 24:145–152
- Dumas Y, Desrosiers J, Soumis F (1991) The pickup and delivery problem with time windows. *Eur J Oper Res* 54:7–22
- Fischetti M, Toth P (1989) An additive bounding procedure for combinatorial optimization problems. *Oper Res* 37:319–328
- Frederickson G, Guan D (1992) Preemptive ensemble motion planning on a tree. *SIAM J Comput* 21:1130–1152
- Frederickson G, Guan D (1993) Nonpreemptive ensemble motion planning on a tree. *J Algorithms* 15:29–60
- Frederickson G, Hecht M, Kim C (1978) Approximation algorithms for some routing problems. *SIAM J Comput* 7:178–193
- Fu L, Rilett LR (1998) Expected shortest paths in dynamic and stochastic traffic networks. *Transport Res Part B* 32:449–516
- Garey MR, Johnson DS (1979) *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco
- Gendreau M, Hertz A, Laporte G (1992) New insertion and postoptimization procedures for the traveling salesman problem. *Oper Res* 40:1086–1094
- Gendreau M, Hertz A, Laporte G (1996) The traveling salesman problem with backhauls. *Comput Oper Res* 23:501–508
- Gendreau M, Hertz A, Laporte G (1997) An approximation algorithm for the traveling salesman problem with backhauls. *Oper Res* 45:639–641
- Gendreau M, Laporte G, Vigo D (1999) Heuristics for the traveling salesman problem with pickup and delivery. *Comput Oper Res* 26:699–714
- Ghaziri H, Osman IH (2003) A neural network algorithm for the traveling salesman problem with backhauls. *Comput Ind Eng* 44:267–281
- Ghaziri H, Osman IH (2006) Self-organizing feature map for the vehicle routing problem with backhauls. *J Sched* 9:97–114
- Glover F (1995) Tabu thresholding: improved search trajectories by non-monotonic search trajectories. *ORSA J Comput* 7:426–442
- Goetschalckx M, Jacobs-Blecha C (1989) The vehicle routing problem with backhauls. *Eur J Oper Res* 42:39–51
- Golden BL, Bodin LD, Doyle T, Stewart WR Jr (1980) Approximate travelling salesman algorithms. *Oper Res* 28:694–711
- Golden BL, Wang Q, Liu L (1988) A multifaceted heuristic for the orienteering problem. *Nav Res Logist* 35:359–366
- Gribkovskaia I, Halskau Ø, Myklebost KNB (2001) Models for the pick-up and deliveries from depots with lasso solutions. In: NOFOMA2001, Collaboration in logistics: connecting islands using information technology, pp 279–293
- Gribkovskaia I, Laporte G, Shlopak A (2006) A tabu search heuristic for a routing problem arising in the servicing of offshore platforms (submitted for publications)
- Gribkovskaia I, Halskau Ø, Laporte G, Vlček M (2007) General solutions to the single vehicle routing problem with pickups and deliveries. *Eur J Oper Res* 180:568–584
- Gélinas S, Desrochers M, Desrosiers J (1995) A new branching strategy for time constrained routing problems with application to backhauling. *Ann Oper Res* 61:91–109
- Halse K (1992) Modeling and solving complex vehicle routing problems. PhD thesis, Technical University of Denmark
- Healy P, Moll R (1980) A new extension of local search applied to the dial-a-ride problem. *Transport Sci* 14:130–154
- Hernández-Pérez H, Salazar-González JJ (2003) The one-commodity pickup-and-delivery traveling salesman problem. In: Jünger M, Reinelt G, Rinaldi G (eds) *Eureka, You shrink!*. Lecture notes in computer science, vol 2570. Springer, Berlin
- Hernández-Pérez H, Salazar-González JJ (2004a) A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discret Appl Math* 145:126–139
- Hernández-Pérez H, Salazar-González JJ (2004b) Heuristics for the one-commodity pickup-and-delivery traveling salesman problem. *Transport Sci* 38:245–255

- Hoff A, Løkketangen A (2006) Creating lasso-solutions for the traveling salesman problem with pickup and delivery by tabu search. *Central Eur J Oper Res* 14:125–140
- Hoff A, Laporte G, Løkketangen A, Gribkovskaia I (2006) Lasso solutions strategies for the vehicle routing problem with pickups and deliveries (submitted for publication)
- Ioachim I, Desrosiers J, Dumas Y, Solomon MM, Villeneuve D (1995) A request clustering algorithm for door-to-door handicapped transportation. *Transport Sci* 29:63–78
- Jaw JJ, Odoni AR, Psaraftis HN, Wilson NHM (1986) A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transport Res Part B* 20:243–257
- Johnson DS, McGeoch LA (1997) The traveling salesman problem: a case study in local optimization. In: Aarts EJJ, Lenstra JK (eds) *Local search in combinatorial optimization*. Wiley, Chichester, pp 215–310
- Kalantari B, Hill AV, Arora SR (1985) An algorithm for the traveling salesman problem with pickup and delivery customers. *Eur J Oper Res* 22:377–386
- Karp RM (1977) Probabilistic analysis of partitioning algorithms for the travelling salesman problem in the plane. *Math Oper Res* 2:209–224
- Katoh N, Yano T (2006) An approximation algorithm for the pickup and delivery vehicle routing problem on trees. *Discret Appl Math* 154:2335–2349
- Kohonen T (1995) Self-organized formation of topologically correct feature maps. *Biol Cybern* 43:59–69
- Kontoravdis G, Bard JF (1995) A GRASP for the vehicle routing problem with time windows. *ORSA J Comput* 7:10–23
- Kruskal JB (1956) On the shortest spanning subtree and the traveling salesman problem. *Proc Am Math Soc* 7:48–50
- Kubo M, Kasugai H (1990) Heuristic algorithms for the single vehicle dial-a-ride problem. *J Oper Res Soc Jpn* 33:354–365
- Lau HC, Liang Z (2002) Pickup and delivery with time windows: algorithms and test case generation. *Int J Artif Intell Tools* 11:455–472
- Lin S (1965) Computer solutions of the traveling salesman problem. *Bell Syst Tech J* 44:2245–2269
- Lin S, Kernighan BW (1973) An effective heuristic algorithm for the traveling salesman problem. *Oper Res* 21:498–516
- Little JDC, Murty KG, Sweeney DW, Karel C (1963) An algorithm for the traveling salesman problem. *Oper Res* 11:972–989
- Lübbecke ME (2004) Combinatorially simple pickup and delivery paths. *Central Eur J Oper Res* 12:405–417
- Madsen OBG, Ravn HF, Rygaard JM (1995) A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Ann Oper Res* 60:193–208
- Mageean J, Nelson JD (2003) The evaluation of demand responsive transport services in Europe. *J Transp Geog* 6:255–270
- Miller C, Tucker A, Zemlin R (1960) Integer programming formulation of traveling salesman problems. *J Assoc Comput Mach* 7:326–329
- Min H (1989) The multiple the vehicle routing problem with simultaneous delivery and pick-up points. *Transport Res Part A* 5:377–386
- Mingozi A, Giorgi S, Baldacci R (1999) An exact method for the vehicle routing problem with backhauls. *Transport Sci* 33:315–329
- Mitrović-Minić S, Laporte G (2004) Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transport Res Part B* 38:635–655
- Mitrović-Minić S, Laporte G (2006) The pickup and delivery problem with time windows and transshipment. *INFOR* 44:217–227
- Mitrović-Minić S, Krishnamurti R, Laporte G (2004) Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transport Res Part B* 38:669–685
- Mladenović N, Hansen P (1997) Variable neighborhood search. *Comput Oper Res* 24:1097–1100
- Montané FAT, Galvão RD (2006) A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Comput Oper Res* 33:595–619
- Mosheiov G (1994) The traveling salesman problem with pick-up and delivery. *Eur J Oper Res* 79:299–310
- Nagy G, Salhi S (2005) Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *Eur J Oper Res* 162:126–141
- Nanry WP, Barnes JW (2000) Solving the pickup and delivery problem with time windows using reactive tabu search. *Transport Res Part B* 34:107–121
- Nowak MA (2005) The pickup and delivery problem with split loads. PhD thesis, Georgia Institute of Technology

- Or I (1976) Traveling Salesman-type combinatorial problems and their relation to the logistics of regional blood banking. PhD thesis, Northwestern University, Evanston, IL
- Osman IH, Wassan NA (2002) A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls. *J Sched* 5:263–285
- Oxley JG (1993) Matroid theory. Oxford University Press, New York
- Pacheco JA (1997) Heurístico para los problemas de rutas con carga y descarga en sistemas LIFO. *Qüestió* 21:153–175
- Padberg MW, Rinaldi G (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Rev* 33:60–100
- Paquette J, Cordeau J-F, Laporte G (2006) Une étude comparative de divers modèles pour le problème de transport à la demande (submitted for publication)
- Peano G (1890) Sur une courbe qui remplit toute une aire plane. *Math Ann* 36:157–160
- Potvin J-Y, Duhamel C, Guertin F (1996) A genetic algorithm for vehicle routing with backhauling. *Appl Intell* 6:345–355
- Potvin J-Y, Rousseau J-M (1992) Constraint-directed search for the advanced-request dial-a-ride problem with service quality constraints. In: Balci RSO, Zenios S (eds) *Computer science and operation research: new developments in their interfaces*. Pergamon, Oxford
- Powell WB, Jaillet P, Odoni AR (2005) Stochastic and dynamic networks and routing. In: Ball MO, Mag-nanti TL, Monma CL, Nemhauser GL (eds) *Network routing. Handbooks in operations research and management science*, vol 8. North-Holland, Amsterdam, pp 141–295
- Privé J, Renaud J, Boctor F, Laporte G (2006) Solving a vehicle-routing problem arising in soft-drink distribution. *J Oper Res Soc* 57:1045–1052
- Psaraftis HN (1980) A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem. *Transport Sci* 14:130–154
- Psaraftis HN (1983a) Analysis of an  $O(n^2)$  heuristic for the single vehicle many-to-many Euclidean dial-a-ride problem. *Transport Res B* 17:133–145
- Psaraftis HN (1983b) An exact algorithm for the single-vehicle, many-to-many dial-a-ride problem with time windows. *Transport Sci* 17:351–357
- Rekiek B, Delchambre A, Saleh HA (2006) Handicapped person transportation: An application of the grouping genetic algorithm. *Eng Appl Artif Intell* 19:511–520
- Robinson RM (1948) Recursion and double recursion. *Bull Am Math Soc* 54:987–993
- Ropke S, Cordeau J-F (2007) Branch-and-cut-and-price for the pickup and delivery problem with time windows (submitted for publication)
- Ropke S, Pisinger D (2006a) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transport Sci* 40:455–472
- Ropke S, Pisinger D (2006b) A unified heuristic for a large class of vehicle routing problems with back-hauls. *Eur J Oper Res* 171:750–775
- Ropke S, Cordeau J-F, Laporte G (2007) Models and a branch-and-cut algorithm for pickup and delivery problems with time windows. *Networks* 49:258–272
- Roy S, Chapleau L, Ferland JA, Lapalme G, Rousseau J-M (1983) The construction of routes and schedules for the transportation of the handicapped. Working paper, Technical Report CRT-308, Université de Montréal
- Ruland KS, Rodin EY (1997) The pickup and delivery problem: faces and branch-and-cut algorithm. *Comput Math Appl* 33:1–13
- Salhi S, Nagy G (1999) A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *J Oper Res Soc* 50:1034–1042
- Salhi S, Rand GK (1987) Improvements to vehicle routing heuristics. *J Oper Res Soc* 38:293–295
- Savelsbergh MWP (1985) Local search in routing problems with time windows. *Ann Oper Res* 4:285–305
- Savelsbergh MWP (1990) An efficient implementation of local search algorithms for constrained routing problems. *Eur J Oper Res* 47:75–85
- Savelsbergh MWP, Sol M (1995) The general pickup and delivery problem. *Transport Sci* 29:17–29
- Savelsbergh MWP, Sol M (1998) DRIVE: dynamic routing of independent vehicles. *Oper Res* 46:474–490
- Sexton TR, Bodin LD (1985a) Optimizing single vehicle many-to-many dial-a-ride problem with desired delivery times, I: scheduling. *Transport Sci* 19:378–410
- Sexton TR, Bodin LD (1985b) Optimizing single vehicle many-to-many dial-a-ride problem with desired delivery times, II: routing. *Transport Sci* 19:411–435
- Stein D (1978) An asymptotic, probabilistic analysis of a routing problem. *Math Oper Res* 3:89–101
- Süral H, Bookbinder JH (2003) The single-vehicle routing problem with unrestricted backhauls. *Networks* 41:127–136

- Thangiah SR, Potvin J-Y, Sun T (1996) Heuristic approaches to vehicle routing with backhauls and time windows. *Comput Oper Res* 23:1043–1057
- Toth P, Vigo D (1996) Fast local search algorithms for the handicapped persons transportation problem. In: Osman H, Kelly J (eds) *Meta-heuristics theory and applications*. Kluwer, Boston, pp 677–690
- Toth P, Vigo D (1997) Heuristic algorithms for the handicapped persons transportation problem. *Transport Sci* 31:60–71
- Toth P, Vigo D (1999) A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. *Eur J Oper Res* 113:528–543
- Toth P, Vigo D (2002) VRP with backhauls. In: Toth P, Vigo D (eds) *The vehicle routing problem*. SIAM, Philadelphia, pp 195–224
- Van der Bruggen LJJ, Lenstra JK, Schuur PC (1993) Variable-depth search for the single-vehicle pickup and delivery problem with time windows. *Transport Sci* 27:298–311
- Voudouris C, Tsang E (1999) Guided local search and its application to the traveling salesman problem. *Eur J Oper Res* 113:469–499
- Wade AC, Salhi S (2002) An investigation into a new class of vehicle routing problems with backhauls. *Omega* 30:479–487
- Wang F, Lim A, Xu Z (2006) The one-commodity pickup and delivery traveling salesman problem on a path or a tree. *Networks* 48:24–35
- Xu H, Chen Z-L, Rajagopal S, Arunapuram S (2003) Solving a practical pickup and delivery problem. *Transport Sci* 37:347–364
- Yano CA, Chan TJ, Richter LK, Cutler T, Murty KG, McGettigan D (1987) Vehicle routing at quality stores. *Interfaces* 17(2):52–63
- Zhong Y, Cole MH (2005) A vehicle routing problem with backhauls and time windows: a guided local search solution. *Transport Res Part E* 41:131–144