Chapter 6

Pickup-and-Delivery Problems for Goods Transportation

Maria Battarra
Jean-François Cordeau
Manuel Iori

6.1 ▪ Introduction

Pickup-and-Delivery Problems (PDPs) constitute an important family of routing problems in which goods or passengers have to be transported from different origins to different destinations. These problems are usually defined on a graph in which vertices represent origins or destinations for the different entities (or *commodities*) to be transported. PDPs can be classified into three main categories according to the type of demand and route structure being considered. In *many-to-many* (M-M) problems, each commodity may have multiple origins and destinations and any location may be the origin or destination of multiple commodities. These problems arise, for example, in the repositioning of inventory between retail stores or in the management of bicycle or car sharing systems. *One-to-many-to-one* (1-M-1) problems are characterized by the presence of some commodities to be delivered from a depot to many customers and of other commodities to be collected at the customers and transported back to the depot. These have applications, for example, in the distribution of beverages and the collection of empty cans and bottles. They also arise in forward and reverse logistics systems where, in addition to delivering new products, one must plan the collection of used, defective, or obsolete products. Finally, in *one-to-one* (1-1) problems, each commodity has a single origin and a single destination between which it must be transported. Typical applications of these problems are less-than-truckload transportation and urban courier operations. The three types of PDPs are illustrated in Figure 6.1, where the square represents the depot and the other vertices are customers.

Like other VRPs, PDPs may also be classified according to the decision framework being considered and to the availability of information. In *static* problems, one assumes that all problem parameters are deterministic and known a priori before vehicle routes are constructed. *Dynamic* problems are characterized by the fact that some of the information required to make decisions is gradually revealed over time and requires the solution

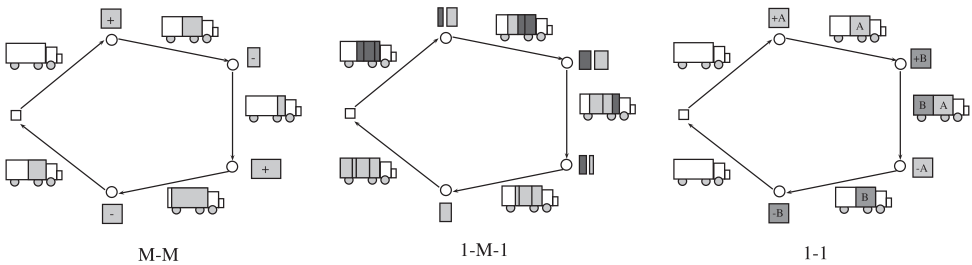

Figure 6.1. *The three types of PDPs.*

to be updated. Dynamic problems may also be *stochastic* when some information about the uncertain parameters is known in the form of probability distributions.

Most variants of PDPs have been studied intensively in the last three decades and have been the object of several literature surveys. In particular, we refer the reader to Parragh, Doerner, and Hartl [103, 104] for a general survey of PDPs and to Berbeglia et al. [15] and Berbeglia, Cordeau, and Laporte [16] for more focused reviews of static and dynamic problems, respectively.

This chapter is concerned with PDPs arising in the transportation of goods. Variants related to passenger transportation are covered in Chapter 7. The next three sections review models and algorithms for many-to-many, one-to-many-to-one, and one-to-one problems, respectively. Problems incorporating vehicle loading constraints are reviewed in Section 6.5.

6.2 ▪ Many-to-Many Problems

Most of the literature on *many-to-many* (M-M) problems focuses on the single-commodity case where a unique commodity must be transported between multiple origins and destinations. The *One-Commodity M-M Pickup and Delivery Vehicle Routing Problem* (1-PDVRP) can be defined as follows. We are given a complete graph $G = (V, A)$ and a fleet K of identical vehicles, each having capacity Q. Vertex 0 is the depot, where vehicles are initially located, and the other vertices of $V \setminus \{0\}$ represent customers. Each customer i has a demand q_i, where $q_i > 0$ means that the customer requires a pickup and $q_i < 0$ that it requires a delivery. Vehicles can leave the depot either empty or with some load. The cost of traveling along an arc $(i, j) \in A$ is c_{ij}. The 1-PDVRP is to find a set of routes that start and finish at the depot, serve all requests without violating the vehicle capacity, and have minimum total cost.

The 1-PDVRP can be modeled as a *Mixed-Integer Linear Program* (MILP) by using binary variables x_{ijk} taking value 1 if arc $(i, j) \in A$ is traversed by vehicle $k \in K$, and 0 otherwise, and non-negative variables f_{ij} indicating the load transported on arc $(i, j) \in A$. The model is the following:

$$\text{(6.1)} \qquad \text{(1-PDVRP)} \quad \text{minimize} \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} c_{ij}x_{ijk}$$

$$\text{(6.2)} \qquad \text{s.t.} \quad \sum_{j\in V}\sum_{k\in K} x_{ijk} = 1 \qquad \forall\, i \in V\setminus\{0\},$$

$$\text{(6.3)} \qquad \sum_{j\in V} x_{ijk} - \sum_{j\in V} x_{jik} = 0 \qquad \forall\, i \in V,\ k \in K,$$

$$0 \le f_{ij} \le Q \sum_{k \in K} x_{ijk} \qquad \forall\, i,j \in V, \tag{6.4}$$

$$\sum_{j \in V} f_{ji} - \sum_{j \in V} f_{ij} = q_i \qquad \forall\, i \in V \setminus \{0\}, \tag{6.5}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \le |S| - 1 \qquad \forall\, S \subseteq V \setminus \{0\},\ S \neq \emptyset,\ k \in K, \tag{6.6}$$

$$x_{ijk} \in \{0,1\} \qquad \forall\, i,j \in V,\ k \in K. \tag{6.7}$$

The objective function requires the minimization of the cost. Constraints (6.2) impose that each customer be visited once. Constraints (6.3) enforce that a vehicle does not end its route at a vertex other than the depot, while constraints (6.4) restrict the flow on each arc to be at most Q. Constraints (6.5) set the flow conservation, and constraints (6.6) avoid the presence of subtours in the solution. Note that in the presence of negative demands constraints (6.6) are necessary because flow conservation is not sufficient to ensure that all routes start and finish at the depot. Note also that it is easy to enrich the above model by considering different weight capacities, or by including time windows or a maximum duration constraint for each route.

As mentioned above, model (6.1)–(6.7) does not impose any restriction on the load that leaves and enters the depot. Vehicles are thus allowed to start their routes carrying some goods, thus possibly visiting delivery customers first. Imposing zero initial and final load may result in a high increase in solution cost. Note also that in the 1-PDVRP description we disregarded customers with zero demand, but these can be easily included in the above model with no further modification in the constraints.

We now describe the main algorithms introduced for the single-vehicle and multiple-vehicle variants of the 1-PDVRP.

6.2.1 ▪ Single Vehicle

The single-vehicle case of the 1-PDVRP was formally introduced by Hernández-Pérez and Salazar-González [59], under the name *One-commodity Pickup and Delivery Traveling Salesman Problem* (1-PDTSP). The authors presented mathematical formulations for the symmetric and asymmetric cases, both based on the use of binary variables x_{ij} taking value one if the edge or arc (i,j) is used, and non-negative variables f_{ij} giving the flow of commodity passing on edge or arc (i,j). They then focused on the solution of the symmetric formulation by Branch-and-Cut. By Benders' decomposition they projected out the flow variables and obtained a pure 0-1 model including an exponential number of Benders' cuts. They proposed an exact procedure to separate these cuts using a maximum flow algorithm on a supporting graph. The Benders' cuts are then lifted to stronger rounded inequalities, which are separated with a four-step heuristic algorithm using in turn two max-flow computations, a constructive greedy procedure and a local search phase. The Branch-and-Cut was enriched by the additional separation of three-edge clique inequalities and a simple primal heuristic invoked at the root node. The proposed formulation was also adapted to solve the 1-M-1 TSP with pickups and deliveries (see Section 6.3.2.1). To test their algorithm they created a set of instances that has become the standard benchmark for comparing algorithms for the 1-PDTSP. These instances have Euclidean distances and a number of requests between 20 and 50. Their Branch-and-Cut could solve to optimality all such instances within two hours of computing time. Computational evidence showed that (i) the 1-PDTSP is harder than the 1-M-1 TSP with pickups and deliveries (see Section 6.3.2.1) as well as the TSP with the same number of vertices;

(ii) the difficulty of the 1-PDTSP increases when the capacity decreases because the vehicle is forced to perform long detours to achieve feasibility and the lower bound is far from the optimum.

Hernández-Pérez and Salazar-González [61] later improved their results with a new Branch-and-Cut algorithm. They again considered the symmetric formulation involving only the x_{ij} binary variables, but enriched it with several families of valid inequalities, namely rank inequalities (including the simpler clique and clique-cluster inequalities), generalized inhomogeneous multistar inequalities (including the simpler positive and negative multistar inequalities), and generalized homogeneous partial multistar inequalities. Most of the separation procedures were based on heuristics and were provided in the form of pseudo-codes. New and existing instances with up to 200 requests were solved within two hours.

To tackle larger instances, Hernández-Pérez and Salazar-González [60] proposed two simple heuristics. The first one is based on a greedy algorithm improved with 2-opt and 3-opt moves, and the second one is based on Branch-and-Cut. The algorithms were tested on small instances ($n \leq 60$) for which optimal solutions are known and on larger instances with $n \leq 500$. On the small instances the optimality gap was always below 2%. The second heuristic provided better solutions than the first. Improved results were obtained by Hernández-Pérez, Rodríguez-Martín, and Salazar-González [58], who introduced a *Variable Neighborhood Descent* (VND) heuristic based on 2-opt and 3-opt local search operators. To diversify the search, the VND was embedded into a GRASP framework. The best solution obtained was post-optimized using a second VND based on vertex exchange operators. On the small instances this heuristic obtained the optimal solution for 145 instances out of 150. For the large instances, by running their algorithm 25 times on each instance they could improve the results of Hernández-Pérez and Salazar-González [60] on 113 of 150 instances. The average CPU time was under 10 minutes.

Zhao et al. [151] obtained good quality solutions by means of a genetic algorithm making use of pheromone information and local search. Their computing times (reported only for the large instances) were comparable to those of Hernández-Pérez, Rodríguez-Martín, and Salazar-González [58]. Hosny and Mumford [64] obtained further improvements by using a *Variable Neighborhood Search* (VNS) algorithm, but at the expense of very high computing times (up to 42 hours on the large instances). To our knowledge, the best computational results have been obtained by Mladenović et al. [92]. They proposed a VNS algorithm that makes use of a collection of neighborhood structures to solve the classical TSP and a binary indexed tree to efficiently check the feasibility of the routes generated during the neighborhoods exploration.

To the best of our knowledge, only one study has addressed the case of multiple commodities: Hernández-Pérez and Salazar-González [62] considered the problem in which a single vehicle is used to transport a set of multiple commodities. They solved the problem with a Branch-and-Cut algorithm similar to that of Hernández-Pérez and Salazar-González [61] for the single-commodity case.

6.2.2 ▪ Multiple Vehicles

The multiple-vehicle case has attracted far less attention than its single-vehicle counterpart. Shi, Zhao, and Gong [125] have presented the mathematical formulation (6.1)–(6.7) and a genetic algorithm derived from that of Zhao et al. [151]. Computational experiments were performed only for the genetic algorithm, on a set of instances derived from those of Hernández-Pérez, Rodríguez-Martín, and Salazar-González [58], by introducing a maximum length constraint for each route. Dror, Fortin, and Roucairol [40] studied the distribution of electric cars in France. Cars are parked in some slots and, after being used, may be returned to a different location. Redistribution is then performed by means of auto-carriers to restore the best distribution for the users. The authors studied the split delivery generalization of the 1-PDVRP, obtained by splitting a vertex of demand q_i into q_i vertices of unit demand. They also considered a heterogeneous fleet of auto-carriers. They modeled the problem with a three-index formulation, similar to (6.1)–(6.7), and also presented heuristic algorithms. Note that this work may be considered as an antecedent of the 1-PDTSP presented by Hernández-Pérez and Salazar-González [59], although it does not report extensive computational results.

A problem that recently attracted the interest of many researchers is the bike sharing rebalancing problem. Bike sharing systems offer public bikes located in several stations. A user can take a bike from a station, use it for a journey, and then leave it at a possibly different station. To keep the system operational, it is often necessary to redistribute the bikes among the stations. Chemla, Meunier, and Wolfler Calvo [27] studied the case of a single vehicle, allowing temporary drop off and split deliveries. Raviv, Tzur, and Forma [111] focused instead on the multiple-vehicle case, proposing a three-index mathematical formulation and testing it on the systems in use in Paris and Washington DC.

6.3 ▪ One-to-Many-to-One Problems

One-to-many-to-one (1-M-1) problems refer to PDPs in which deliveries and pickups concern two distinct sets of commodities: some are shipped from the depot to the customers, and others are picked up at the customers and delivered to the depot. In the last two decades, 1-M-1 problems have become increasingly popular, mainly because of the trend toward recycling and product reuse. For example, in Europe the delivery of new appliances should be accompanied by the pickup of old ones in order to comply with Waste Electrical and Electronic Equipment (WEEE) regulations (see, e.g., Beullens, Van Oudheusden, and Van Wassenhove [17]). The beverage industry provides another popular area of application with full bottles being delivered and empty ones being returned to the depot (see, e.g., Privé et al. [107]). There are also applications in which loaded pallets or containers have to be delivered to the customers while empty ones are collected from the customers (see, e.g., Crainic, Gendreau, and Dejax [35]).

The literature on 1-M-1 PDPs is vast and varied. It is therefore convenient to further classify its variants with respect to additional features concerning the customer ordering, the demand types, and the presence of time windows or maximum route duration constraints. Similar classifications have been proposed for the single-vehicle variants in the survey of Gribkovskaia and Laporte [55]. In particular, problems can be classified with respect to the type of service considered: customers may ask for a *simultaneous* (or combined) service of pickup and delivery, when at least one customer requires both a pickup and a delivery. *Single-demand* service occurs when every customer requires either a pickup or a delivery, but not both.

1-M-1 problems are also intrinsically connected with handling issues: pickups on board may obstruct deliveries. Situations in which handling operations are not allowed may force the decision maker to serve all deliveries first, followed by all pickups. In such a case, if a customer requires both services, it will have to be visited twice. These problems are referred to as *backhaul* problems. On the contrary, problems in which handling is acceptable are referred to as *mixed*, because routes allow for mixed sequences of pickups and deliveries. Intermediate situations are those in which a subset of customers (or all of them) are allowed to be visited more than once. The resulting routes are not Hamiltonian tours, but may allow one to avoid handling operations at the expense of an increase in routing cost. Variants in which handling operations are avoided by altering the Hamiltonian tour shape are presented in Section 6.5.

In Figure 6.2, single-vehicle problems with simultaneous and single demand, backhauls, and mixed services are depicted. Vertex 0 is the depot, and each customer i requires d_i units of delivery from the depot and provides p_i units to be picked up (denoted by the pair (d_i, p_i)). Cases (c) and (d) differ from (a) and (b) because customer 1 requires a combined pickup and delivery service of 2 delivery units and 1 pickup. Tour (a) differs from tour (b) with respect to the backhaul constraint: in tour (b) the delivery customer 2 has to be served before the others (requiring pickup), resulting in a higher routing cost. In tour (d), customer 1 has to be visited twice: first to perform the delivery and second to perform the pickup, thus respecting the backhaul constraint.

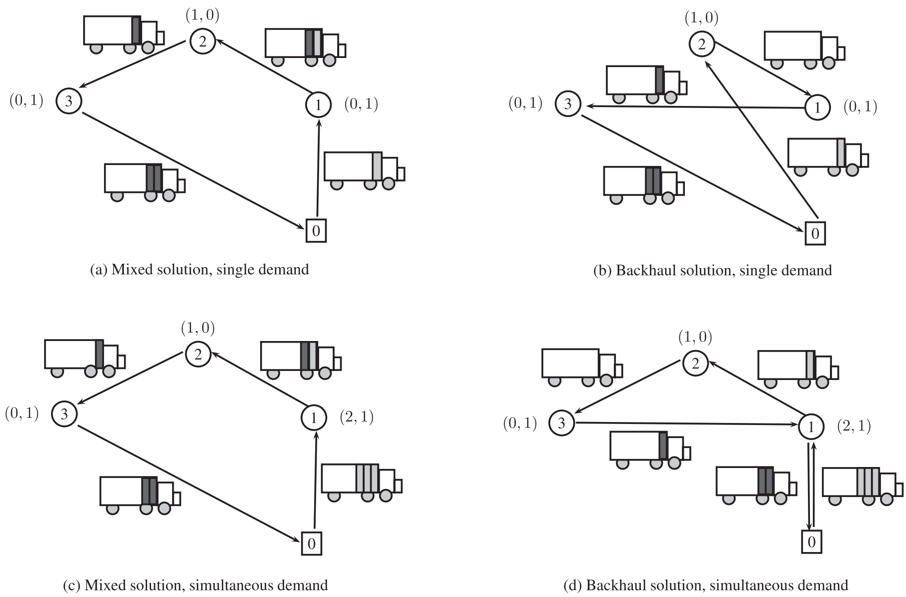

Figure 6.2. *Different* 1-*M*-1 *problems.*

In the remainder of this section, problems with combined and single demands as well as problems with mixed routes are presented, in that order.

6.3.1 ▪ Simultaneous Demands

The *Vehicle Routing Problem with Simultaneous Pickup and Delivery* (VRPSPD) is probably the most studied and most general variant of the 1-M-1 problems, and it is also known as the *Multiple-Vehicle Hamiltonian* 1-*M*-1-*PDP with Combined Demands*. It can be formally defined as follows. Let $G = (V, A)$ be a directed graph where V is the vertex set and A is the arc set. Vertex 0 represents the depot, which acts as the origin of the delivery commodities and destination of the pickup commodities, and the other vertices of V are the customers. We assume that a homogeneous fleet K of vehicles with capacity Q is available. As for the 1-PDVRP, the cost of traversing the arc (i, j) is c_{ij}. Each customer i has non-negative demands d_i for the delivery commodity and p_i for the pickup commodity. The problem is to construct $|K|$ routes of minimum total cost, satisfying the pickup and delivery requests of each customer in a single visit and not exceeding the capacity of the vehicles.

The following is an adaptation of the three-index MIP formulation for the VRPSPD of Montané and Galvão [93]. In this formulation, x_{ijk} takes value 1 if the arc (i,j) is traversed by vehicle k, and 0 otherwise. The flow variables y_{ij} and z_{ij} are the amounts of pickup and delivery commodities traveling on arc (i,j), respectively:

$$\text{(VRPSPD)} \quad \text{minimize} \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} c_{ij}x_{ijk} \tag{6.8}$$

$$\text{s.t.} \quad \sum_{j\in V}\sum_{k\in K} x_{ijk} = 1 \qquad \forall\, i \in V\setminus\{0\}, \tag{6.9}$$

$$\sum_{j\in V} x_{ijk} - \sum_{j\in V} x_{jik} = 0 \qquad \forall\, i\in V,\ k\in K, \tag{6.10}$$

$$\sum_{i\in V} x_{0ik} \le 1 \qquad \forall\, k\in K, \tag{6.11}$$

$$y_{ij} + z_{ij} \le Q\sum_{k\in K} x_{ijk} \qquad \forall\, i,j\in V, \tag{6.12}$$

$$\sum_{j\in V} y_{ij} - \sum_{j\in V} y_{ji} = p_i \qquad \forall\, i\in V\setminus\{0\}, \tag{6.13}$$

$$\sum_{j\in V} z_{ji} - \sum_{j\in V} z_{ij} = d_i \qquad \forall\, i\in V\setminus\{0\}, \tag{6.14}$$

$$y_{ij}, z_{ij} \ge 0 \qquad \forall\, i,j\in V, \tag{6.15}$$

$$x_{ijk} \in \{0,1\} \qquad \forall\, i,j\in V,\ k\in K. \tag{6.16}$$

The objective function (6.8) aims at minimizing the routing cost. Constraints (6.9) and (6.10) force each customer to be visited exactly once. Constraints (6.11) impose that each vehicle in the fleet be used at most once, while (6.12) bound the flow of goods traveling on each arc to be at most equal to the vehicle capacity. Constraints (6.13) and (6.14) are the flow conservation constraints for pickups and deliveries, respectively. Note that this formulation does not consider additional constraints such as time windows (see Angelelli and Mansini [4]) or maximum route duration constraints (see Montané and Galvão [93], Ai and Kachitvichyanukul [1], and Mingyong and Erbao [89]).

We now present a summary of the main algorithms for the single-vehicle and multiple-vehicle cases, respectively.

6.3.1.1 ▪ Single Vehicle

The single-vehicle problem, which is usually called the *TSP with Simultaneous Pickup and Delivery* (TSPSPD) or *Single-Vehicle Hamiltonian* 1-*M*-1-*PDP with Combined Demands*, asks for the minimum-cost Hamiltonian tour serving each customer once without violating the capacity of the vehicle. The TSPSPD can be transformed into a problem with single demands (see, e.g., Berbeglia et al. [15]) by setting the vehicle capacity equal to $Q' = Q - \sum_{i\in V\setminus\{0\}} d_i$ and defining the demand of customer i as $\delta_i = p_i - d_i$ for all $i \in V\setminus\{0\}$. Obviously, the single-demand variant is also a special case of the TSPSPD. Therefore, algorithms for the combined and single-demand problems are interchangeable (when the tour is Hamiltonian). Most of the research has focused on the case with single demands. We thus refer the reader to Section 6.3.2.1 for algorithms that can be adapted to the TSPSPD. We finally note that it is also possible to solve the TSPSPD as a 1-PDTSP, as mentioned by Hernández-Pérez and Salazar-González [59].

6.3.1.2 ▪ Multiple Vehicles

The multiple-vehicle Hamiltonian problem is often called the *VRP with Simultaneous Pickup and Delivery* (VRPSPD) and has attracted a lot of attention in the scientific community. The VRPSPD was introduced by Min [87], inspired by the problem of distributing and collecting books in a library. He proposed a three-stage constructive heuristic in which customers are first clustered, then vehicles are assigned to clusters, and finally routes are assigned to the vehicles. Anily [5] proposed a lower bound and a heuristic algorithm that is asymptotically optimal, under mild probabilistic conditions on the customer locations.

Among early heuristics, one may mention the insertion-based heuristic of Salhi and Nagy [120], able to solve problems with single and multiple depots, the cheapest insertion algorithm with quadratic complexity insertion criteria of Dethloff [39], the cluster-first-route-second algorithm of Halse [56], and the local search heuristic of Nagy and Salhi [96], in which different degrees of solution feasibility are accepted during the execution.

Several of the metaheuristics proposed to solve the VRPSPD are based on *Tabu Search* (TS) or are hybrid algorithms borrowing the concept of tabu list from TS. Crispim and Brandão [36] proposed a TS algorithm in which the neighborhood exploration is performed according to a VND procedure and a single tabu list keeps track of forbidden moves for all neighborhoods. Chen and Wu [28] proposed another hybrid metaheuristic, where a record-to-record algorithm is enhanced by adopting a tabu list and classical local search operators. Montané and Galvão [93] presented a TS algorithm, as well as a mathematical formulation for the VRPSPD with maximum route duration constraints. Bianchessi and Righini [18] enhanced the TS approach by a VND scheme for the neighborhood selection. We also mention the reactive TS algorithms of Wassan, Nagy, and Ahmadi [143] and Wassan, Wassan, and Nagy [144] and the hybrid TS and guided local search algorithm of Zachariadis, Tarantilis, and Kiranoudis [148].

Among successful population-based metaheuristics, Çatay [26] introduced an effective ant colony algorithm. Zachariadis, Tarantilis, and Kiranoudis [149] proposed an adaptive memory approach, in which genetic algorithms and TS are hybridized. More precisely, promising solution components, or *bones*, are extracted from a pool of routes. Bones are selected according to a utility function, combining the bone's length, the number of times the bone was selected in previous iterations, and the cost of the best solution in which the bone was present. Moreover, bones are selected so that each customer appears only once, allowing for an easy recombination of the selected bones into a feasible VRPSPD solution. A granular TS algorithm is applied to improve the resulting solution, and the pool of routes is updated before the whole algorithm is iterated. Finally, Zachariadis and Kiranoudis [147] proposed a local search metaheuristic exploring 2-opt and variable length bone exchange neighborhoods. The latter aims at exchanging the positions of sequences of customers of length shorter than μ. A static move descriptor strategy allows for a faster exploration of these rich neighborhoods, and the "promises" concept enhances the algorithm by preventing cycling and by inducing diversification.

Successful and flexible single-solution-based metaheuristics for the VRPSPD are the *Adaptive Large Neighborhood Search* (ALNS) of Ropke and Pisinger [117], the *Parallel Iterated Local Search* (PILS) algorithm of Subramanian et al. [131], and the Unified local search and Hybrid Genetic Search (UHGS) algorithm of Vidal et al. [140]. ALNS iteratively perturbs and improves a solution by means of many diversification and intensification neighborhoods. The neighborhoods are chosen at each iteration by a roulette wheel selection principle, in which weights are adaptively updated according to their performance in previous iterations. The PILS algorithm consists of an intensification stage, in which neighborhoods with quadratic complexity are exhaustively explored in random order, and a diversification stage, in which a randomly selected rich neighborhood perturbs the current solution. The candidate diversification neighborhoods are double-bridge, double swap, and ejection chains. The algorithm is initialized by a randomized constructive heuristic and implemented in a master-slave parallel architecture. A similar *Iterated Local Search* (ILS) hybrid algorithm was presented in a non-parallel implementation by Subramanian, Uchoa, and Ochi [132]. This algorithm consists of a cooperative framework, in which an ILS algorithm similar to that of Subramanian et al. [131] and a *Set Partitioning* (SP) formulation share a pool of routes and upper bounds. The ILS algorithm provides high quality upper bounds and columns in reasonable computing time to the SP formulation, whereas the ILS algorithm benefits from high-quality initial solutions yielded by the SP formulation. The resulting framework, denoted by H-ILS-SP, provides high quality results not only when applied to the VRPSPD, but to many routing variants. The UHGS of Vidal et al. [140] is based on a hybrid genetic algorithm, in which local search operators improve individuals in the population, benefiting from a solution representation without trip delimiters. The individual evaluation preserves (but penalizes) infeasible solutions and diverse individuals; the selection strategy promotes diversity and cost efficiency by keeping track of subpopulations (each updated with respect to the survivor selection mechanism, when the size is too large).

The most common benchmark instances for the VRPSPD are the real-world instance of Min [87] (with 22 customers), the instances of Dethloff [39] (40 problems with 50 customers), the set by Salhi and Nagy [120] (14 problems with 50 to 199 customers), and the instances of Montané and Galvão [93] (18 problems with 100 to 400 customers). The latter two sets are probably the most challenging arena for metaheuristics: the best results seem to be achieved by the adaptive memory algorithm of Zachariadis, Tarantilis, and Kiranoudis [149], the PILS, the H-ILS-SP, and the UHGS. More precisely, UHGS and PILS algorithms obtain a 0.00% average percentage deviation from the best known solutions for the instances of Salhi and Nagy [120], whereas the average gap for Zachariadis, Tarantilis, and Kiranoudis [149] is 0.11%. Note that PILS is considerably faster than UHGS (given the parallel implementation) but provides a slightly worse average performance. The best results on the benchmark set of Montané and Galvão [93] are achieved by H-ILS-SP, with 0.00% and 0.08% average gaps for the best and the average solution values among the 10 executions being performed. UHGS produces slightly worse results (0.07% and 0.20% average gaps for the best and the average of 10 replications).

To the best of our knowledge, the only approximation algorithm for the VRPSPD is that of Katoh and Yano [69], where the networks being analyzed are trees.

Exact algorithms include the Branch-and-Cut-and-Price algorithm for the VRPSPD with time windows by Angelelli and Mansini [4], capable of solving to optimality instances with up to 20 customers, and the algorithm of Dell'Amico, Righini, and Salani [37]. This algorithm is based on a set covering formulation where exact dynamic programming and state space relaxation techniques are used to solve the pricing subproblem. Bi-directional search, an upper bound on the number of customers visited in a route, as well as problem-specific branching strategies allowed the algorithm to solve to optimality problems with up to 40 customers.

Recently, Subramanian et al. [133] proposed a Branch-and-Cut algorithm based on the lazy separation of the capacity inequalities, improving the best known lower bounds and finding some new optimal solutions. This approach seems to outperform the Branch-and-Cut algorithms based on single-commodity, undirected two-commodity, and directed two-commodity formulations (capable of solving instances with up to 100 customers) from Subramanian [129]. Subramanian et al. [134] also proposed a Branch-and-Cut-and-

Price method based on the *pd*-routes idea: *pd*-routes are non-elementary paths in which both pickups and deliveries are taken into account to identify capacity violations. The pricing problem is solved by dynamic programming, generating *pd*-routes, but scaling and sparsification are adopted at early stages of the algorithm to speed up computation. This algorithm has improved many best known lower bounds and seems to outperform Branch-and-Cut algorithms when the average number of customers per route is small.

6.3.2 ▪ Single Demands

Single-demand 1-M-1 problems represent a special case of simultaneous demand problems where each customer can either require a pickup or a delivery, but not both. These problems can be further classified depending on the constraints imposed on the customer order: mixed routes allow pickup and delivery customers to be mixed (see Section 6.3.2.1 for single-vehicle problems and Section 6.3.2.2 for multiple-vehicle problems), whereas backhaul routes force all delivery customers to precede the pickups. Backhaul problems will not be presented extensively in this chapter but are covered in Chapter 9.

6.3.2.1 ▪ Single Vehicle

In problems with mixed routes, deliveries and pickups can be performed in any order. As mentioned by Gribkovskaia and Laporte [55], the single-vehicle problem is usually referred to as the *TSP with Pickups and Deliveries* (TSPPD), the *TSP with Delivery and Backhauls*, and the *Mixed TSP*. In the classification of Berbeglia et al. [15], the problem is the 1-*M*-1-*PDP with Single Demands and Mixed Solutions*. In the following, we will adopt the acronym TSPPD.

The TSPPD was introduced by Mosheiov [94], who proved that any TSP solution can be converted into a feasible TSPPD solution by inserting the depot in an appropriate location along the tour. Assuming that the TSP tour is obtained by an algorithm with worst-case performance α, the overall approach guarantees a worst-case ratio of $1+\alpha$. Mosheiov [94] also described a constructive heuristic based on a cheapest insertion criterion. Anily and Mosheiov [6] proposed an approximation algorithm based on the construction of a minimum spanning tree: the tree is traversed in a depth-first fashion, prioritizing the subtrees with positive net demand and delivery customers in the same subtree. Finally, a Hamiltonian tour is obtained by means of shortcut operations. Assuming symmetric distances satisfying the triangular inequality, the worst-case performance of this algorithm is 2.

Gendreau, Laporte, and Vigo [51] considered the special case of a TSPPD defined on a cycle and designed a linear-time exact algorithm. The algorithm was then adapted to general graphs and provides a worst-case performance equal to 3. The authors also proposed a TS metaheuristic, based on 2-opt moves, and a set of instances that now constitute the standard benchmark for the TSPPD. These instances were adapted from VRP instances, and the number of customers varies between 6 and 261.

Some effective algorithms for the M-M problem have also been extended to the TSPPD, such as the Hernández-Pérez and Salazar-González truncated Branch-and-Cut algorithm for the 1-PDTSP [60] (see Section 6.2.1). Another adaptation to the TSPPD is the hybrid genetic algorithm of Zhao et al. [152] proposed for the 1-PDTSP: the algorithm is based on a pheromone-driven crossover operator and local search operators. Recently, Çinar, Öncan, and Haldun [29] proposed another genetic algorithm using a depot removal-insertion-based tour improvement procedure.

The ILS of Subramanian and Battarra [130] combines a VND intensification phase (in which Or-opt, 2-opt, exchange, and relocate operators are applied in random order) with an effective diversification phase (in which the double-bridge operator is used), similar to the ILS metaheuristic for the VRPSPD described in Section 6.3.1.2.

Finally, Baldacci, Hadjiconstantinou, and Mingozzi [10] proposed a Branch-and-Cut algorithm, based on a two-commodity flow formulation, with initial upper bounds obtained by a genetic algorithm. This approach could solve to optimality some instances with up to 200 customers. The state-of-the-art exact algorithm for the TSPPD is an adaptation of the Branch-and-Cut of Hernández-Pérez and Salazar-González [59, 61], presented in Section 6.2.1. This algorithm is able to solve instances with up to 260 customers.

Classical benchmark problems for the TSPPD are the Euclidean instances of Mosheiov [94] and the instances of Gendreau, Laporte, and Vigo [51]. Both sets are available at http://webpages.ull.es/users/hhperez/PDsite/. The Gendreau, Laporte, and Vigo set [51] consists of instances adapted from the VRP (6 to 261 customers), instances in which the coordinates of the customers are randomly generated and Euclidean instances are considered (25 to 200 customers), and instances with randomly generated arc costs (25 to 200 customers). Gendreau, Laporte, and Vigo [51], Hernández-Pérez and Salazar-González [60], Zhao et al. [152], and Çinar, Öncan, and Haldun [29] report high-quality average performance for the Gendreau, Laporte, and Vigo problems [51]. Note that the genetic algorithm used by Baldacci, Hadjiconstantinou, and Mingozzi [10] to compute upper bounds had also found optimal solutions to all instances in the set but seven. The randomly generated instances of Hernández-Pérez and Salazar-González [60] provide a more challenging benchmark set involving up to 500 customers. The algorithms of Hernández-Pérez and Salazar-González [60] and of Subramanian and Battarra [130] produce the best known results: both are capable of solving to optimality all small instances (up to 60 customers) in fractions of a second of computing time. The former is slightly faster than the latter on large instances (100 to 500 customers), but the latter, despite its simplicity, was capable of finding 44 new best known solutions and provides a better performance than the former, even by considering its average performance over 10 executions.

6.3.2.2 ▪ Multiple Vehicles

The *Vehicle Routing Problem with Mixed Pickup and Delivery* (VRPMPD) or *Multiple-Vehicle* 1-*M*-1 *PDP with Single Demands and Mixed Solutions* was first studied by Golden et al. [53] under the name *Vehicle Routing with Backhauling*. The authors introduced a stop-based backhaul procedure according to which delivery customers are scheduled first, while pickups are added to the routes in a second stage by an insertion algorithm. Casco, Golden, and Wasil [25] enhanced this idea by incorporating a penalty term in the insertion procedure that is applied whenever deliveries are performed after pickups.

Mosheiov [95] proposed a two-index formulation, a lower bounding procedure, and iterative tour partitioning algorithms with related worst-case performance for the VRPMPD. These heuristics iteratively break a giant tour into routes, trying to optimize the utilization of the vehicles. The constructive heuristics of Salhi and Nagy [120] and of Dethloff [39] (presented in Section 6.3.1.2) are capable of solving the VRPMPD, and so are the ALNS of Ropke and Pisinger [117], the local search heuristic of Nagy and Salhi [96], and the hybrid TS and VND algorithm by Crispim and Brandão [36]. Wade and Salhi [142] proposed an ant colony algorithm dedicated to the VRPMPD, while Wassan, Nagy, and Ahmadi [143] discussed the relationship between the simultaneous and mixed variants and developed another ant colony algorithm. We refer the interested reader to Wassan, Nagy, and Ahmadi [143] for a description of five sets of benchmark instances.

Subramanian [129] presented both exact and heuristic algorithms for the VRPMPD. The exact algorithms are Branch-and-Cut algorithms based on lazy separation of the capacity constraints (see also Subramanian et al. [133]), on single-commodity, undirected two-commodity, directed two-commodity, and two-index formulations, as well as on the Branch-and-Cut-and-Price algorithm presented in Section 6.3.1.2 (see also Subramanian et al. [134]). These algorithms provide the best lower bounds on the Salhi and Nagy benchmark set [120] (21 instances without route duration constraint and 21 with route duration constraint, considering between 50 and 199 customers). The Branch-and-Cut-and-Price algorithm improves some of the best known lower bounds and outperforms the Branch-and-Cut algorithms, given that the average number of customers per route in the Salhi and Nagy problems [120] is relatively small. The ILS of Subramanian [129] and the hybrid algorithm of Subramanian, Uchoa, and Ochi [132] (presented in Section 6.3.1.2) are also capable of solving the VRPMPD: both algorithms outperform the ALNS of Ropke and Pisinger [116]. The hybrid algorithm is capable of finding or improving the best known solutions on all instances but a few. The ILS is not as accurate as the hybrid, but about six times faster.

Kontoravdis and Bard [70] presented a GRASP algorithm for the VRP with time windows and adapted the algorithm for the VRPMDP with time windows. Zhong and Cole [153] proposed a guided local search heuristic for the same variant. Wade and Salhi [141] studied a VRPMDP variant in which pickups may be performed only once a percentage of the deliveries has been delivered. This problem is solved by means of an insertion heuristic. Finally, good computational results were recently obtained by Tarantilis, Anagnostopoulou, and Repoussis [136]. They studied the effect of different backhauling strategies, varying from the pure mixed strategy to the one allowing only backhaul routes (see the next section). They proposed an adaptive path relinking method that evolves a set of reference solutions and combines them using path relinking. Their algorithm obtained good improvements with respect to existing metaheuristics for the VRPMDP with time windows.

6.3.2.3 ▪ Backhaul Routes

In problems with backhaul routes, all deliveries must be performed before any of the pickups. Therefore, handling issues are always avoided at the expense of extra routing costs. We refer the reader to the survey of Gribkovskaia and Laporte [55] and to Chapter 9 for the single-vehicle and multiple-vehicle problems, respectively.

Another family of related problems arises in the context of transportation with *cross-docking* (see, e.g., Wen et al. [145] and Tarantilis [135]), where vehicles performing pickups stop at a consolidation facility to reorganize the loads before making deliveries. Recently, Santos, Mateus, and da Cunha [121] explicitly considered cross-docking in pickup and delivery routing, allowing for both routes that visit pickup locations followed by delivery locations and routes that stop at an intermediate cross-dock.

6.4 ▪ One-to-One Problems

In *one-to-one* (1-1) PDPs, each customer request consists of transporting a load from one pickup vertex to one destination vertex. This problem arises in urban courier services, less-than-truckload transportation systems and, maritime shipping (see Andersson, Christiansen, and Fagerholt [3] and Korsvik, Fagerholt, and Laporte [71]), among others. In the context of passenger transportation, it is often called the dial-a-ride problem (see Chapter 7).

We first provide a formulation for the general case involving multiple capacitated vehicles and time windows. We then review the literature on the single-vehicle case and on the multiple-vehicle case, respectively.

Following the notation introduced by Cordeau [30], 1-1 problems can be formulated on a graph $G = (V,A)$ where the vertex set V is the union of the set of pickup nodes

$P = \{1,\dots,n\}$, the set of delivery nodes $D = \{n+1,\dots,2n\}$, and the depot nodes $\{0, 2n+1\}$. Each request i then consists of transporting a load of size q_i from the pickup node $i \in P$ to the delivery node $n+i \in D$. A non-negative service duration s_i is also associated with every node $i \in V$. We assume that $s_0 = s_{2n+1} = 0$, $q_{n+i} = -q_i$, and $q_0 = q_{2n+1} = 0$. A time window $[a_i, b_i]$ is also associated with node $i \in V$, where a_i and b_i represent the earliest and latest times, respectively, at which service may begin at node i. Finally, with each arc $(i, j) \in A$ are associated a routing cost c_{ij} and a travel time t_{ij}. Let L_k be the maximum duration of the route performed by vehicle k.

For each arc $(i, j) \in A$ and each vehicle $k \in K$, let x_{ijk} be a binary variable equal to 1 if vehicle k travels from node i to node j, and 0 otherwise. For each node $i \in V$ and each vehicle $k \in K$, let T_{ik} be the time at which vehicle k begins service at node i, and let Q_{ik} be the load of vehicle k after visiting node i. The *Pickup-and-Delivery Vehicle Routing Problem with Time Windows* (PDVRPTW) can be formulated as the following mixed-integer program:

$$(6.17)\quad \text{(PDVRPTW)} \quad \text{minimize} \sum_{k\in K}\sum_{i\in V}\sum_{j\in V} c_{ij}x_{ijk}$$

$$(6.18)\quad \text{s.t.} \quad \sum_{k\in K}\sum_{j\in V} x_{ijk} = 1 \qquad \forall\, i \in P,$$

$$(6.19)\quad \sum_{j\in V} x_{ijk} - \sum_{j\in V} x_{n+i,jk} = 0 \qquad \forall\, i \in P,\ k \in K,$$

$$(6.20)\quad \sum_{j\in V} x_{0jk} = 1 \qquad \forall\, k \in K,$$

$$(6.21)\quad \sum_{j\in V} x_{jik} - \sum_{j\in V} x_{ijk} = 0 \qquad \forall\, i \in P \cup D,\ k \in K,$$

$$(6.22)\quad \sum_{i\in V} x_{i,2n+1,k} = 1 \qquad \forall\, k \in K,$$

$$(6.23)\quad T_{jk} \geq (T_{ik} + s_i + t_{ij})x_{ijk} \qquad \forall\, i \in V,\ j \in V,\ k \in K,$$

$$(6.24)\quad Q_{jk} \geq (Q_{ik} + q_j)x_{ijk} \qquad \forall\, i \in V,\ j \in V,\ k \in K,$$

$$(6.25)\quad T_{n+i,k} - T_{ik} - s_i - t_{i,n+i} \geq 0 \qquad \forall\, i \in P,$$

$$(6.26)\quad T_{2n+1,k} - T_{0k} \leq L_k \qquad \forall\, k \in K,$$

$$(6.27)\quad a_i \leq T_{ik} \leq b_i \qquad \forall\, i \in V,\ k \in K,$$

$$(6.28)\quad \max\{0, q_i\} \leq Q_{ik} \leq \min\{Q_k, Q_k + q_i\} \qquad \forall\, i \in V,\ k \in K,$$

$$(6.29)\quad x_{ijk} \in \{0,1\} \qquad \forall\, i \in V,\ j \in V,\ k \in K.$$

Constraints (6.18) and (6.19) ensure that each request is served exactly once and that the origin and destination nodes of a request are visited by the same vehicle. Constraints (6.20)–(6.22) guarantee that all routes start at the origin depot and end at the destination depot. Consistency of the time and load variables is ensured by constraints (6.23) and (6.24). Precedence constraints are imposed through inequalities (6.25). Finally, inequalities (6.26) bound the duration of each route, while (6.27) and (6.28) impose time-window and capacity constraints, respectively.

6.4.1 ▪ Single Vehicle

Most of the research on single-vehicle problems has focused on variants without time windows. The special case of the problem where the load of each request is equal to 1 and the vehicle itself has a capacity of 1 is called the *stacker crane problem*. This problem, which has applications in port operations and in various industrial contexts, was studied, among others, by Frederickson, Hecht, and Kim [49].

The more general *Pickup-and-Delivery TSP* (PDTSP) was first studied formally in the context of passenger transportation by Stein [127, 128], who performed a probabilistic analysis of the problem and introduced construction heuristics based on region-partitioning principles. Several dynamic programming approaches were then developed for the PDTSP. In particular, Psaraftis [108] and Desrosiers, Dumas, and Soumis [38] addressed problems arising in dial-a-ride systems.

The incorporation of precedence constraints in models and algorithms for the TSP was first discussed by Lokin [80]. Later, Kalantari, Hill, and Arora [68] extended the TSP Branch-and-Bound algorithm of Little et al. [79] to solve the PDTSP with a single or multiple vehicles and with or without capacity constraints. A polyhedral study of the PDTSP was done by Ruland and Rodin [118], who introduced a Branch-and-Cut algorithm based on four families of valid inequalities. This algorithm was capable of solving instances with up to 15 requests. This work was later extended by Dumitrescu et al. [42], who generalized some of these inequalities and introduced several other families. Their Branch-and-Cut algorithm was able to solve some instances with up to 35 requests.

The development of exact algorithms for the PDTSP has largely benefited from the work done on the closely related TSP with precedence constraints in which a vertex may have multiple predecessors. The polyhedron of this problem was analyzed by Balas, Fischetti, and Pulleyblank [8], whereas Fischetti and Toth [48] and Ascheuer, Jünger, and Reinelt [7] solved the problem by additive Branch-and-Bound and by Branch-and-Cut, respectively. Dynamic programming approaches were also proposed by Bianco et al. [19] and Mingozzi, Bianco, and Ricciardelli [88]. Finally, additional valid inequalities were described by Gouveia and Pesneau [54] and by Mak and Ernst [85].

Local search heuristics for the PDTSP were first investigated by Psaraftis [109], who considered k-interchanges as in the Lin and Kernighan heuristic [78] for the TSP. The author proposed a procedure to identify the best such move in $O(n^k)$ despite the presence of the precedence constraints. The idea consists of separating the improvement checks, i.e., finding the best move, from the feasibility checks, i.e., verifying its feasibility. Feasibility checks can be performed in a preprocessing step, and the information can be stored in a matrix which is updated every time an improved solution is found. Later, Kubo and Kasugai [72] introduced several insertion heuristics along with two local search algorithms extending the Or-opt procedure for the TSP by Or [99]. The efficient implementation of local search operators for the TSP with precedence constraints was also addressed by Savelsbergh [122]. A variable-depth search procedure relying on seven types of arc exchange mechanisms was developed by Van der Bruggen, Lenstra, and Schuur [138] to solve the PDTSP with time windows. An improvement to local search heuristics for the PDTSP was also introduced by Healy and Moll [57], who suggested alternating between moves aiming to improve a solution in the traditional sense and moves bringing the search toward solutions that have larger feasible neighborhoods. This principle, called "sacrificing", can be seen as a type of algorithmic perturbation. Several other heuristics for the PDTSP were introduced by Renaud, Boctor, and Ouenniche [113] and Renaud, Boctor, and Laporte [112]. The most effective of these heuristics are based on solution perturbation ideas. Vertices can be randomly removed and reinserted in the solution, or two solutions can be combined into one as in crossover operations performed in genetic algorithms. Finally, a genetic algorithm, a simulated annealing approach, and a hill climbing heuristic were developed by Hosny and Mumford [65] to solve the PDTSP with time windows.

6.4.2 ▪ Multiple Vehicles

A large number of exact and heuristic algorithms were developed for the multiple-vehicle 1-to-1 problem, which is usually referred to as the *Pickup-and-Delivery Vehicle Routing Problem* (PDVRP). The vast majority of these algorithms handle the more general problem with time windows.

A Branch-and-Cut algorithm for both the PDVRP and the PDVRPTW was introduced by Lu and Dessouky [81], who formulated the problem with a polynomial number of variables and constraints through the use of binary variables b_{ij} indicating whether node i comes before another node j on the tour. The formulation was also strengthened by the addition of four families of valid inequalities relying on these variables. The authors were able to solve instances with up to five vehicles and 25 requests. Another Branch-and-Cut algorithm was developed by Ropke, Cordeau, and Laporte [115], who focused on the PDVRPTW and introduced two formulations with an exponential number of constraints based on the precedence constraints of Ruland and Rodin [118]. The most compact formulation in terms of the number of variables uses only binary routing variables, and time-window constraints are imposed by forbidding infeasible paths. These formulations were improved by several families of valid inequalities, including strengthened capacity constraints that take the pickup-and-delivery structure into account and fork constraints that combine this structure with time-window restrictions. The Branch-and-Cut algorithm was able to solve some instances with up to 75 requests.

Several algorithms based on column generation were also proposed for the PDVRPTW. The first such algorithm was introduced by Dumas, Desrosiers, and Soumis [41], who formulated the problem as an SP problem with an extra constraint on the number of vehicles. Columns are generated by solving through forward dynamic programming a non-elementary shortest path problem with pickup, delivery, and time-window constraints. In their implementation of the shortest path algorithm, labels contain the set of open requests, i.e., requests for which the pickup node has been visited, but not the delivery. This algorithm was capable of solving tightly constrained instances with up to 55 requests. Later, Ropke and Cordeau [114] presented a Branch-and-Cut-and-Price algorithm incorporating some of the inequalities introduced by Ropke, Cordeau, and Laporte [115] in a Branch-and-Price framework. The authors have compared solving the elementary and non-elementary shortest path problems to generate columns. This algorithm outperformed the Branch-and-Cut algorithm of Ropke, Cordeau, and Laporte [115] and was able to solve some tightly constrained instances with up to 500 requests. Very recently, another exact algorithm based on column generation was introduced by Baldacci, Bartolini, and Mingozzi [9]. This algorithm relies on a bounding procedure that finds a near-optimal dual solution of the LP relaxation of the SP formulation by combining two dual ascent heuristics and a cut-and-column generation procedure. This solution is then used to enumerate the routes with a reduced cost smaller than the gap between the lower and upper bounds. If the number of columns is small enough, an optimal solution can be found by solving an integer program. Otherwise, Branch-and-Cut-and-Price is used to close the gap. This algorithm is much faster than that of Ropke and Cordeau [114] and was able to solve 15 more instances to optimality.

One of the first heuristics for the PDVRPTW was introduced by Nanry and Barnes [97], who developed a reactive TS relying on three move operators: moving a pickup-delivery pair from its current route to a different route, swapping two pickup-delivery pairs between two different routes, and moving an individual pickup or delivery node within its route. An improved implementation of this heuristic was later described by Lau and Liang [74], who also proposed a hybrid partitioned insertion heuristic to generate initial solutions. At approximately the same time, Li and Lim [76] proposed another heuristic based on very similar move operators. This heuristic combines TS and simulated annealing and forces the search to return to the incumbent solution after a number of iterations without improvement. Pankratz [102] proposed a grouping genetic algorithm in which each gene represents a group of requests instead of single requests. An insertion heuristic is used to update the routing corresponding to a group of requests whenever an individual is modified by a genetic operator. Lu and Dessouky [82] focused on the creation of initial solutions and introduced an insertion-based construction heuristic that differs from classical procedures by taking into consideration the increase in travel time as well as the reduction in time window slack due to the insertion. In addition, this heuristic tries to reduce the number of crossings in the constructed routes so as to improve the visual attractiveness of the solution.

The two best performing metaheuristics for the PDVRPTW are those of Bent and Van Hentenryck [14] and of Ropke and Pisinger [116], both based on large neighborhood search. The first is a two-stage hybrid algorithm that consists of a first stage aiming to decrease the number of routes by using simulated annealing and a second stage trying to reduce the total travel cost by using large neighborhood search. This algorithm produced very good solutions on a large number of test instances. However, improved solutions were obtained slightly later by the ALNS of Ropke and Pisinger [116]. This heuristic relies on multiple simple heuristics both to remove and to reinsert customers in the solution. To remove nodes from the solution, the authors consider an adaptation of the relatedness measure of Shaw [124], as well as additional rules such as worst removal, which removes the requests leading to the largest cost savings. Reinsertions are performed either via a greedy or a regret heuristic. On test problems involving up to 1000 requests, this heuristic has produced better results on average than the two-stage approach of Bent and Van Hentenryck [14], both in terms of the number of vehicles used and of the total distance traveled.

While most algorithms developed for the PDVRP consider the static variant of the problem where all requests are known in advance, several authors have also studied the dynamic case in which some requests arrive dynamically as other requests are already being served. Savelsbergh and Sol [123] developed a Branch-and-Price algorithm to address a problem arising at a road transportation company operating in the Benelux region. The algorithm was used heuristically to solve large-scale dynamic instances over a multiple-day planning horizon by decomposing them into a number of smaller subproblems containing known requests within a rolling horizon framework. Mitrović-Minić and Laporte [91] defined and compared four waiting strategies to decide where vehicles should wait when requests arrive dynamically in the presence of time windows on pickups and deliveries. The authors considered a rolling horizon in which static problems are solved by a cheapest insertion procedure followed by a TS. This approach was then extended by Mitrović-Minić, Krishnamurti, and Laporte [90] by considering a double horizon objective: a short term objective of minimizing distance and a long term objective of maximizing slack time to accommodate new requests. A different TS heuristic for this problem was developed by Gendreau et al. [50]. This heuristic relies on ejection chain neighborhoods and an adaptive memory that maintains a pool of routes extracted from the best visited solutions. A hybrid adaptive and predictive control algorithm combining genetic algorithms and fuzzy clustering was introduced by Sáez, Cortés, and Núñez [119] for a dynamic PDVRP. The algorithm takes into account future demand and prediction of expected waiting time and travel times for the requests. Finally, Ghiani et al. [52] have described and assessed anticipatory heuristics for the problem that take advantage of stochastic information about future demand in the hope of making better vehicle assignment, routing, and waiting decisions. The algorithms work by sampling the short-term future every time a new request arrives so as to take into account the inconvenience cost of future requests when evaluating potential solutions. The authors have devised both an insertion procedure and a local search algorithm, which they applied to instances containing up to 600 requests. Both algorithms significantly outperformed reactive algorithms that do not incorporate sampling.

Several other variants of the PDVRP have also been studied in recent years. One such variant is the PDVRP with split loads that was addressed by Nowak, Ergun, and White [98]. The authors quantified the benefits of allowing split loads in the context of the PDP and conjectured that load splitting can lead to cost reductions of at most 50%. They have also proposed a simple heuristic to generate an initial solution and then select loads for splitting according to some guidelines. On a related note, Venkateshan and Mathur [139] designed a specialized column generation algorithm for a variant of the PDVRP where multiple visits are required by the same or different vehicles to satisfy each request. Another interesting variant is the PDVRP with transshipment that was recently introduced by Qu and Bard [110] in the context of air freight transportation. A GRASP incorporating a reactive ALNS heuristic was proposed to solve the problem. This heuristic makes use of specialized insertion and removal procedures to accommodate transshipments.

Several industrial applications of the PDVRP have also been described. For example, Xu et al. [146] addressed a rich PDP with many side constraints and features: multiple carriers and vehicle types, multiple time windows, compatibility constraints between requests and vehicles as well as between requests, last-in-first-out vehicle loading policy, working hour rules for drivers, and a complex objective function. This model aims to capture most practical situations arising at the client firms of an optimization software provider. The authors have proposed solving the problem with a column generation–based approach that solves the LP relaxation to optimality and then obtains an integer solution by considering only the set of columns generated when solving this relaxation. An application concerning the transportation of live animals was studied by Sigurd, Pisinger, and Sig [126]. In this context, precedence constraints must be enforced to avoid the spread of diseases, with healthier animals being transported first. The authors formulated the problem as a PDVRPTW with precedence constraints and developed a column generation approach to solve it to optimality.

6.5 ▪ Problems with Loading Constraints

In several pickup-and-delivery applications, the nature of the freight to be transported and the characteristics of the vehicle and of the handling equipment may cause some loading or unloading operations to be costly or even impossible. This happens, for example, when the vehicle is rear-loaded (i.e., it has a single door at the back for loading and unloading), and items are heavy or fragile and hence cannot be rearranged at the customer locations. In this situation an item being picked up is placed at the end of the cargo, and the next item to be delivered must be taken only from those at the back of the vehicle. Consider, for example, the 1-1 problem depicted in Figure 6.1. After the two pickups of A and B, the item available at the end of the cargo is B, and hence the first delivery to be performed can only be that of B, followed by A. This typically leads to an increase in solution cost with respect to problems where loading issues need not be taken into account.

Loading considerations are usually modeled by imposing additional constraints in the model, by including handling costs in the objective function to represent the time lost (or risk incurred) for a cargo rearrangement, or by reducing the loading space available on the vehicle so as to allow any possible loading or unloading operation. Several real-world problems involving combined routing (not necessarily with pickup and delivery) and loading have been studied in the literature. We refer the reader to Iori and Martello [66] for a recent survey. In this section we focus on 1-1 and 1-M-1 PDPs involving loading constraints. To the best of our knowledge, these constraints have not been considered in the context of M-M problems.

6.5.1 ▪ One-to-One Problems

An obvious situation where pickup and delivery operations are affected by loading considerations is when the shapes of the items complicate their arrangement in the cargo and must thus be explicitly taken into account when planning vehicle routes.

Malapert et al. [86] studied the generalization of the PDVRP (see Section 6.4.2), in which items are two-dimensional rectangles and vehicles are rear-loaded and have a two-dimensional rectangular loading surface. This problem also generalizes the *Capacitated Vehicle Routing Problem* (CVRP) with two-dimensional loading constraints (see, e.g., Iori, Salazar-González, and Vigo [67]) by introducing pickup and delivery of the items. A pickup of an item may be performed only if there is a feasible non-overlapping placement for the item in the vehicle surface. A delivery may be performed only if the unloading of the item can be obtained with a single straight movement, as would happen when using an automatic fork. No reshuffling of the cargo is allowed. The resulting problem was modeled using constraint programming, but no results were given.

Bartók and Imreh [11] studied a generalization of the above problem, where vehicles have a three-dimensional loading space and items are three-dimensional boxes. They solved the problem using heuristics based on local search operators. Zachariadis, Tarantilis, and Kiranoudis [150] investigated a further generalization in which time windows are imposed and the three-dimensional items are first loaded on pallets, which are then loaded on vehicles.

6.5.1.1 ▪ TSP with Pickup and Delivery and LIFO or FIFO Loading

The natural extension of the PDTSP (see Section 6.4.1) in which the vehicle is rear-loaded and the loading or unloading operations must be performed according to a *Last-In-First-Out* (LIFO) policy is known in the literature as the *Traveling Salesman Problem with Pickup and Delivery and LIFO Loading* (TSPPDL). The first work related to this problem dates back to Ladany and Mehrez [73], who studied a real-life delivery problem. Then, Pacheco [100, 101] proposed a Branch-and-Bound and a heuristic algorithm based on Or-opt exchanges.

More recently, heuristic algorithms were developed by Carrabs, Cordeau, and Laporte [23], who proposed exchange operators and a variable neighborhood search heuristic, and by Li et al. [77], who presented a metaheuristic based on a tree representation of the problem. Carrabs, Cerulli, and Cordeau [21] implemented an exact Branch-and-Bound using additive lower bounds and consistently solved instances with up to 17 requests. Cordeau et al. [32] solved the problem with a Branch-and-Cut based on the classical two-index vehicle flow formulation and consistently solved instances with up to 21 requests. By looking at the results, it is obvious that the TSPPDL is computationally more difficult than the original PDTSP.

Generalizations of the TSPPDL involving multiple vehicles were studied by Levitin and Abezgaouz [75], who focused on the routing of multiple-load LIFO AGVs in industrial plants, and by Xu et al. [146], who solved a real-world delivery problem with several complicating constraints (see Section 6.4.2 for more details).

A variant of the TSPPDL that considers the *First-In-First-Out* (FIFO) rule, instead of the LIFO, has been studied under the name of *Traveling Salesman Problem with Pickup and Delivery and FIFO Loading* (TSPPDF). Erdoğan, Cordeau, and Laporte [43] proposed a multi-commodity ILP formulation and several heuristics, including TS and iterated local search. Carrabs, Cerulli, and Cordeau [21] solved the problem by Branch-and-Bound, and Cordeau, Dell'Amico, and Iori [31] used Branch-and-Cut. As for the TSPPDL, the TSPPDF is hard and some instances with just 15 requests are still not solved to proven optimality.

6.5.1.2 ▪ Double TSP with Multiple Stacks

A variant that recently attracted the interest of many researchers is the *Double Traveling Salesman Problem with Multiple Stacks* (DTSPMS), another generalization of the PDTSP discussed in Section 6.4.1. The main difference between the two problems is the fact that in the DTSPMS each customer request consists of one pallet, and the single vehicle is equipped with multiple stacks, each of which accommodates pallets by obeying the LIFO policy. The vehicle must perform a first Hamiltonian tour on the pickup vertices and then a second Hamiltonian tour on the delivery vertices. During the pickup phase each pallet is loaded at the top of a stack, and during the delivery phase only the pallets located at the top of a stack can be unloaded. The aim is to find a pair of cycles of minimum cost for which there exists a feasible loading and unloading plan.

The DTSPMS was motivated by a real-world distribution case and was first studied by Petersen and Madsen [106], who proposed an ILP formulation and a simulated annealing metaheuristic. In terms of exact algorithms, Lusby et al. [84] presented an algorithm based on the iterated execution of two phases: in the first phase, the k best solutions of the two separate TSPs are generated; in the second phase, attempts are made to find feasible loading plans for each pair of TSP solutions. The approach fails in solving some instances with 12 requests. It was later improved by Lusby and Larsen [83], who proposed pre-processing techniques and increased the size of the smallest unsolved instance to 14 requests. An enumerative Branch-and-Bound for the case in which the vehicle has exactly two stacks was developed by Carrabs, Cerulli, and Speranza [22].

The best results on the DTSPMS have been obtained by Branch-and-Cut. Petersen, Archetti, and Speranza [105] studied mathematical formulations with exponentially many constraints and solved them with Branch-and-Cut algorithms. Alba Martínez et al. [2] considered one of the formulations of Petersen, Archetti, and Speranza [105] and enriched it with several families of valid inequalities. Their Branch-and-Cut could solve to optimality all instances with 18 requests. The fact that instances with about 20 requests remain unsolved after several attempts is an indication of the combinatorial difficulty of the DTSPMS.

To address larger instances, heuristic algorithms have been proposed. Felipe, Ortuño, and Tirado [45, 46] presented neighborhood structures and derived a variable neighborhood search algorithm. This last algorithm was extended by Felipe, Ortuño, and Tirado [47] with the use of algorithms that also search in the space of infeasible solutions. The complexity of the DTSPMS and of several subproblems that may arise from it is discussed by Toulouse and Wolfler Calvo [137], Casazza, Ceselli, and Nunkesser [24], and Bonomo, Mattia, and Oriolo [20].

The generalization of the DTSPMS in which pickups and deliveries may be performed in mixed order has been studied by Côté, Gendreau, and Potvin [34], who solved it with a large neighborhood search procedure, and by Côté et al. [33], who proposed mathematical formulations and Branch-and-Cut algorithms. Batista-Galván, Riera-Ledesma, and

Salazar-González [12] proposed a Branch-and-Cut algorithm and several valid inequalities for a further relaxation of the DTSPMS in which delivery vertices represent customers requiring a given product, whereas pickup vertices are markets where products are available at different costs. The problem is to find the minimum cost tour that visits a subset of markets, loads the products into the stacks, and then delivers the products to the customers.

6.5.2 ▪ One-to-Many-to-One Problems

In some cases the loading of the vehicle is organized in such a way that all loading and unloading operations are possible at any moment along the route. This, however, is obtained at the expense of a reduction in the overall loading capacity of the vehicle. This practice is applied, for example, to overcome linehaul/backhaul distribution and save routing costs. For example, Hoff et al. [63] studied a real-world beverage distribution problem arising in Norway. This is a typical situation of the 1-M-1 transportation discussed in Section 6.3, where full bottles must be delivered and empty bottles collected. In their study, each vehicle is rear-loaded, and its initial cargo is organized so that all full bottles are packed along the sides of the vehicle. An empty corridor is left at the center of the cargo, making all bottles accessible for unloading. Empty bottles, when loaded, take the place of the full ones that have been unloaded.

In other cases the full loading space of the vehicle can be used, but in order to perform unloading operations some items may have to be reshuffled at the customer sites. In this case the additional effort required for the reshuffling is usually modeled as a handling cost. A typical example is given by the *Traveling Salesman Problem with Pickups, Deliveries, and Handling Costs* (TSPPD-H). The TSPPD-H generalizes the TSP with Simultaneous Pickup and Delivery (see Section 6.3.1.1) by adding handling costs for all those unloading operations that cannot be performed directly from the bottom of a single rear-loaded vehicle. The problem was introduced by Battarra et al. [13], who modeled it as an ILP. Since several possible rearrangements of a cargo may be performed after a delivery, the authors introduced some simplified policies to make the problem more tractable and proposed heuristics and a Branch-and-Cut algorithm. These algorithms were tested on instances involving up to 25 customers and hundreds of items. Larger instances were solved by Erdoǧan et al. [44] using metaheuristic algorithms. They first considered the subproblem of computing the optimal rearrangement of a cargo once the vehicle route is fixed. They then proposed an exact and an approximate method to solve this subproblem, and used them within TS and iterated local search algorithms.

6.6 ▪ Conclusions and Future Research Directions

This chapter has described the main variants of PDPs that have been studied in the last two decades. It has also reviewed the main exact and heuristic algorithms developed to solve these problems. One can notice that there is an abundant literature concerning each of the three main problem variants: many-to-many, one-to-many-to-one, and one-to-one transportation. There is also a growing interest in problems that combine pickup and delivery routing with vehicle loading considerations.

While the focus has for a long time been on heuristics to handle large-scale instances, one now witnesses an increasing number of exact algorithms capable of solving instances of realistic size. Nevertheless, most PDP instances remain much harder to solve than classical VRP instances of the same size. This is largely due to the presence of the precedence constraints which typically lead to poor linear programming relaxations. More research is thus needed on the polyhedral study of PDPs. In the current state of the art, it seems fair to say that methods based on Branch-and-Price appear to be the most promising for problems with multiple vehicles.

When looking at the pickup and delivery literature as a whole, one cannot fail to notice that there exists a very large number of problem variants which differ in their structure but nevertheless share many similarities. One can thus hope to see the development of general modeling and solution techniques capable of handling multiple variants with a unified framework. This would also parallel the evolution observed in other fields of vehicle routing.

This survey has also highlighted the fact that the literature for multiple-vehicle many-to-many problems is relatively scarce with respect to the single-vehicle counterparts. Loading features have also mainly been considered in single-vehicle PDPs. Therefore, multiple-vehicle many-to-many problems and multiple-vehicle problems with loading features still leave ample scope for further research opportunities

Finally, as with other routing problems, one observes a gradual trend toward the inclusion of dynamic and stochastic aspects. Most applications of PDPs are dynamic in nature and call for the frequent update of solutions as new information becomes available. With the current availability of information technologies that allow real-time communication between drivers and dispatchers, one can expect that there will be a growing need for algorithms capable of reacting to incoming requests and other changing conditions that require quick decision making.

Bibliography

[1] T. J. AI AND V. KACHITVICHYANUKUL, *A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery*, Computers & Operations Research, 36 (2009), pp. 1693–1702.

[2] M. A. ALBA MARTÍNEZ, J.-F. CORDEAU, M. DELL'AMICO, AND M. IORI, *A branch-and-cut algorithm for the double traveling salesman problem with multiple stacks*, INFORMS Journal on Computing, 25 (2013), pp. 41–55.

[3] H. ANDERSSON, M. CHRISTIANSEN, AND K. FAGERHOLT, *The maritime pickup and delivery problem with time windows and split loads*, INFOR, 49 (2011), pp. 79–91.

[4] E. ANGELELLI AND R. MANSINI, *The vehicle routing problem with time windows and simultaneous pick-up and delivery*, in Quantitative Approaches to Distribution Logistics and Supply Chain Management, Lecture Notes in Economics and Mathematical Systems, A. Klose, M. G. Speranza, and L. N. Van Wassenhove, eds., Springer-Verlag, Berlin, 2001, pp. 249–267.

[5] S. ANILY, *The vehicle-routing problem with delivery and back-haul options*, Naval Research Logistics, 43 (1996), pp. 415–434.

[6] S. ANILY AND G. MOSHEIOV, *The traveling salesman problem with delivery and backhauls*, Operations Research Letters, 16 (1994), pp. 11–18.

[7] N. ASCHEUER, M. JÜNGER, AND G. REINELT, *A branch & cut algorithm for the asymmetric traveling salesman problem with precedence constraints*, Computational Optimization and Applications, 17 (2000), pp. 61–84.

[8] E. Balas, M. Fischetti, and W. R. Pulleyblank, *The precedence-constrained asymmetric traveling salesman polytope*, Mathematical Programming, 68 (1995), pp. 241–265.

[9] R. Baldacci, E. Bartolini, and A. Mingozzi, *An exact algorithm for the pickup and delivery problem with time windows*, Operations Research, 59 (2011), pp. 414–426.

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

[11] T. Bartók and C. Imreh, *Pickup and delivery vehicle routing with multidimensional loading constraints*, Acta Cybernetica, 20 (2011), pp. 17–33.

[12] M. Batista-Galván, J.-J. Riera-Ledesma, and J.-J. Salazar-González, *The traveling purchaser problem, with multiple stacks and deliveries: A branch-and-cut approach*, Computers & Operations Research, 40 (2013), pp. 2103–2115.

[13] M. Battarra, G. Erdoğan, G. Laporte, and D. Vigo, *The traveling salesman problem with pickups, deliveries, and handling costs*, Transportation Science, 44 (2010), pp. 383–399.

[14] R. Bent and P. V. Hentenryck, *A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows*, Computers & Operations Research, 33 (2006), pp. 875–893.

[15] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, *Static pickup and delivery problems: A classification scheme and survey*, TOP, 15 (2007), pp. 1–31.

[16] G. Berbeglia, J.-F. Cordeau, and G. Laporte, *Dynamic pickup and delivery problems*, European Journal of Operational Research, 202 (2010), pp. 8–15.

[17] P. Beullens, D. Van Oudheusden, and L. N. Van Wassenhove, *Collection and vehicle routing issues in reverse logistics*, in Reverse Logistics: Quantitative Models for Closed-Loop Supply Chains, R. Dekker, M. Fleischmann, K. Inderfurth, and L. Van Wassenhove, eds., Springer-Verlag, Heidelberg, 2004, pp. 95–134.

[18] N. Bianchessi and G. Righini, *Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery*, Computers & Operations Research, 34 (2007), pp. 578–594.

[19] L. Bianco, A. Mingozzi, S. Ricciardelli, and M. Spadoni, *Exact and heuristic procedures for the traveling salesman problem with precedence constraints, based on dynamic programming*, INFOR, 32 (1994), pp. 19–31.

[20] F. Bonomo, S. Mattia, and G. Oriolo, *Bounded coloring of co-comparability graphs and the pickup and delivery tour combination problem*, Theoretical Computer Science, 412 (2011), pp. 6261–6268.

[21] F. Carrabs, R. Cerulli, and J.-F. Cordeau, *An additive branch-and-bound algorithm for the pickup and delivery traveling salesman problem with LIFO or FIFO loading*, INFOR, 45 (2007), pp. 223–238.

[22] F. CARRABS, R. CERULLI, AND M. G. SPERANZA, *A branch-and-bound algorithm for the double TSP with two stacks*, Networks, 61 (2013), pp. 58–65.

[23] F. CARRABS, J.-F. CORDEAU, AND G. LAPORTE, *Variable neighbourhood search for the pickup and delivery traveling salesman problem with LIFO loading*, INFORMS Journal on Computing, 19 (2007), pp. 618–632.

[24] M. CASAZZA, A. CESELLI, AND M. NUNKESSER, *Efficient algorithms for the double travelling salesman problem with multiple stacks*, Computers & Operations Research, 39 (2012), pp. 1044–1053.

[25] D. O. CASCO, B. L. GOLDEN, AND E. A. WASIL, *Vehicle routing with backhauls: Models, algorithms, and case studies*, in Vehicle Routing: Methods and Studies, B. L. Golden, ed., North–Holland, Amsterdam, 1988, pp. 127–147.

[26] B. ÇATAY, *A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery*, Expert Systems with Applications, 37 (2010), pp. 6809–6817.

[27] D. CHEMLA, F. MEUNIER, AND R. WOLFLER CALVO, *Bike sharing systems: Solving the static rebalancing problem*, Discrete Optimization, 10 (2013), pp. 120–146.

[28] J.-F. CHEN AND T.-H. WU, *Vehicle routing problem with simultaneous deliveries and pickups*, Journal of the Operational Research Society, 57 (2006), pp. 579–587.

[29] V. ÇINAR, T. ÖNCAN, AND S. HALDUN, *A genetic algorithm for the traveling salesman problem with pickup and delivery using depot removal and insertion moves*, in Applications of Evolutionary Computation, C. Di Chio, A. Brabazon, G. Di Caro, M. Ebner, M. Farooq, A. Fink, J. Grahl, G. Greenfield, P. Machado, M. O'Neill, E. Tarantino, and N. Urquhart, eds., vol. 6025 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2010, pp. 431–440.

[30] J.-F. CORDEAU, *A branch-and-cut algorithm for the dial-a-ride problem*, Operations Research, 54 (2006), pp. 573–586.

[31] J.-F. CORDEAU, M. DELL'AMICO, AND M. IORI, *Branch-and-cut for the pickup and delivery traveling salesman problem with FIFO loading*, Computers & Operations Research, 37 (2010), pp. 970–980.

[32] J.-F. CORDEAU, M. IORI, G. LAPORTE, AND J.-J. SALAZAR-GONZÁLEZ, *A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with LIFO loading*, Networks, 55 (2010), pp. 46–59.

[33] J.-F. CÔTÉ, C. ARCHETTI, M. G. SPERANZA, M. GENDREAU, AND J.-Y. POTVIN, *A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with multiple stacks*, Networks, 60 (2012), pp. 212–226.

[34] J.-F. CÔTÉ, M. GENDREAU, AND J.-Y. POTVIN, *Large neighborhood search for the single vehicle pickup and delivery problem with multiple loading stacks*, Networks, 60 (2012), pp. 19–30.

[35] T. G. CRAINIC, M. GENDREAU, AND P. DEJAX, *Dynamic and stochastic models for the allocation of empty containers*, Operations Research, 41 (1993), pp. 102–126.

[36] J. CRISPIM AND J. BRANDÃO, *Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls*, Journal of the Operational Research Society, 56 (2005), pp. 1296–1302.

[37] M. DELL'AMICO, G. RIGHINI, AND M. SALANI, *A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection*, Transportation Science, 40 (2006), pp. 235–247.

[38] J. DESROSIERS, Y. DUMAS, AND F. SOUMIS, *A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows*, American Journal of Mathematical and Management Sciences, 6 (1986), pp. 301–325.

[39] J. DETHLOFF, *Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up*, OR Spectrum, 23 (2001), pp. 79–96.

[40] M. DROR, D. FORTIN, AND C. ROUCAIROL, *Redistribution of self-service electric cars: A case of pickup and delivery*, Technical Report 3543, INRIA, Institut National de Rercherche en Informatique et en Automatique, Rocquencourt, France, 1998.

[41] Y. DUMAS, J. DESROSIERS AND F. SOUMIS, *The pickup and delivery problem with time windows*, European Journal of Operational Research, 54 (1991), pp. 7–22.

[42] I. DUMITRESCU, S. ROPKE, J.-F. CORDEAU, AND G. LAPORTE, *The traveling salesman problem with pickup and delivery: Polyhedral results and a branch-and-cut algorithm*, Mathematical Programming, 121 (2010), pp. 269–305.

[43] G. ERDOĞAN, J.-F. CORDEAU, AND G. LAPORTE, *The pickup and delivery traveling salesman problem with first-in-first-out loading*, Computers & Operations Research, 36 (2009), pp. 1800–1808.

[44] G. ERDOĞAN, M. BATTARRA, G. LAPORTE, AND D. VIGO, *Metaheuristics for the traveling salesman problem with pickups, deliveries and handling costs*, Computers & Operations Research, 39 (2012), pp. 1074–1086.

[45] A. FELIPE, M. T. ORTUÑO, AND G. TIRADO, *The double traveling salesman problem with multiple stacks: A variable neighborhood search approach*, Computers & Operations Research, 36 (2009), pp. 2983–2993.

[46] ——, *New neighborhood structures for the double traveling salesman problem with multiple stacks*, TOP, 17 (2009), pp. 190–213.

[47] ——, *Using intermediate non-feasible solutions to approach vehicle routing problems with precedence and loading constraints*, European Journal of Operational Research, 211 (2011), pp. 66–75.

[48] M. FISCHETTI AND P. TOTH, *An additive bounding procedure for combinatorial optimization problems*, Operations Research, 37 (1989), pp. 319–328.

[49] G. N. FREDERICKSON, M. S. HECHT, AND C. E. KIM, *Approximation algorithms for some routing problems*, SIAM Journal on Computing, 7 (1978), pp. 178–193.

[50] M. GENDREAU, F. GUERTIN, J.-Y. POTVIN, AND R. SÉGUIN, *Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries*, Transportation Research Part C: Emerging Technologies, 14 (1998), pp. 157–174.

[51] M. GENDREAU, G. LAPORTE, AND D. VIGO, *Heuristics for the traveling salesman problem with pickup and delivery*, Computers & Operations Research, 26 (1999), pp. 699–714.

[52] G. GHIANI, E. MANNI, A. QUARANTA, AND C. TRIKI, *Anticipatory algorithms for same-day courier dispatching*, Transportation Research Part E: Logistics and Transportation Review, 45 (2009), pp. 96–106.

[53] B. L. GOLDEN, E. K. BAKER, J. L. ALFARO, AND J. R. SCHAFFER, *The vehicle routing problem with backhauling*, in Proceedings of the Twenty-First Annual Meeting of the S. E. TIMS, Myrtle Beach, SC, 1985, pp. 90–92.

[54] L. GOUVEIA AND P. PESNEAU, *On extended formulations for the precedence constrained asymmetric traveling salesman problem*, Networks, 48 (2006), pp. 77–89.

[55] I. GRIBKOVSKAIA AND G. LAPORTE, *One-to-many-to-one single vehicle pickup and delivery problems*, in The Vehicle Routing Problem: Latest Advances and New Challenges, B. L. Golden, S. Raghavan, E. A. Wasil, R. Sharda, and S. Voss, eds., vol. 43 of Operations Research/Computer Science Interface Series, Springer-Verlag, Berlin, 2008, pp. 359–377.

[56] K. HALSE, *Modeling and solving complex vehicle routing problems*, PhD thesis, Institute of Mathematical Statistics and Operations Research, Technical University of Denmark, Lyngby, Denmark, 1992.

[57] P. HEALY AND R. MOLL, *A new extension of local search applied to the dial-a-ride problem*, European Journal of Operational Research, 83 (1995), pp. 83–104.

[58] H. HERNÁNDEZ-PÉREZ, I. RODRÍGUEZ-MARTÍN, AND J.-J. SALAZAR-GONZÁLEZ, *A hybrid GRASP/VND heuristic for the one-commodity pickup-and-delivery traveling salesman problem*, Computers & Operations Research, 36 (2009), pp. 1639–1645.

[59] H. HERNÁNDEZ-PÉREZ AND J.-J. SALAZAR-GONZÁLEZ, *A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery*, Discrete Applied Mathematics, 145 (2004), pp. 126–139.

[60] ——, *Heuristics for the one-commodity pickup-and-delivery traveling salesman problem*, Transportation Science, 38 (2004), pp. 245–255.

[61] ——, *The one-commodity pickup-and-delivery traveling salesman problem: Inequalities and algorithms*, Networks, 50 (2007), pp. 258–272.

[62] ——, *The multi-commodity pickup-and-delivery traveling salesman problem*, Networks, 63 (2014), pp. 46–59.

[63] A. HOFF, I. GRIBKOVSKAIA, G. LAPORTE, AND A. LØKKETANGEN, *Lasso solution strategies for the vehicle routing problem with pickups and deliveries*, European Journal of Operational Research, 192 (2009), pp. 755–766.

[64] M. HOSNY AND C. MUMFORD, *Solving the one-commodity pickup and delivery problem using an adaptive hybrid VNS/SA approach*, in Parallel Problem Solving from Nature, PPSN XI, R. Schaefer, C. Cotta, J. Kolodziej, and G. Rudolph, eds., vol. 6239 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2010, pp. 189–198.

[65] ——, *The single vehicle pickup and delivery problem with time windows: Intelligent operators for heuristic and metaheuristic algorithms*, Journal of Heuristics, 16 (2010), pp. 417–439.

[66] M. IORI AND S. MARTELLO, *Routing problems with loading constraints*, TOP, 18 (2010), pp. 4–27.

[67] M. IORI, J.-J. SALAZAR-GONZÁLEZ, AND D. VIGO, *An exact approach for the vehicle routing problem with two-dimensional loading constraints*, Transportation Science, 41 (2007), pp. 253–264.

[68] B. KALANTARI, A. V. HILL, AND S. R. ARORA, *An algorithm for the traveling salesman problem with pickup and delivery customers*, European Journal of Operational Research, 22 (1985), pp. 377–386.

[69] N. KATOH AND T. YANO, *An approximation algorithm for the pickup and delivery vehicle routing problem on trees*, Discrete Applied Mathematics, 154 (2006), pp. 2335–2349.

[70] G. KONTORAVDIS AND J. F. BARD, *A GRASP for the vehicle routing problem with time windows*, ORSA Journal on Computing, 7 (1995), pp. 10–23.

[71] J. E. KORSVIK, K. FAGERHOLT, AND G. LAPORTE, *A large neighbourhood search heuristic for ship routing and scheduling with split loads*, Computers & Operations Research, 38 (2011), pp. 474–483.

[72] M. KUBO AND H. KASUGAI, *Heuristic algorithms for the single vehicle dial-a-ride problem*, Journal of the Operations Research Society of Japan, 33 (1990), pp. 354–365.

[73] S. P. LADANY AND A. MEHREZ, *Optimal routing of a single vehicle with loading constraints*, Transportation Planning and Technology, 8 (1984), pp. 301–306.

[74] H. C. LAU AND Z. LIANG, *Pickup and delivery with time windows: Algorithms and test case generation*, International Journal on Artificial Intelligence Tools, 11 (2002), pp. 455–472.

[75] K. LEVITIN AND R. ABEZGAOUZ, *Optimal routing of multiple-load AGV subject to LIFO loading constraints*, Computers & Operations Research, 30 (2003), pp. 397–410.

[76] H. LI AND A. LIM, *A metaheuristic for the pickup and delivery problem with time windows*, International Journal on Artificial Intelligence Tools, 12 (2003), pp. 173–186.

[77] Y. LI, A. LIM, W.-C. OON, H. QIN, AND D. TU, *The tree representation for the pickup and delivery traveling salesman problem with LIFO loading*, European Journal of Operational Research, 212 (2011), pp. 482–496.

[78] S. LIN AND B. W. KERNIGHAN, *An effective heuristic algorithm for the traveling-salesman problem*, Operations Research, 21 (1973), pp. 498–516.

[79] J. D. C. LITTLE, K. G. MURTY, D. W. SWEENEY, AND C. KAREL, *An algorithm for the traveling salesman problem*, Operations Research, 11 (1963), pp. 972–989.

[80] F. C. J. LOKIN, *Procedures for travelling salesman problems with additional constraints*, European Journal of Operational Research, 3 (1979), pp. 135–141.

[81] Q. LU AND M. M. DESSOUKY, *An exact algorithm for the multiple vehicle pickup and delivery problem*, Transportation Science, 38 (2004), pp. 503–514.

[82] ——, *A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows*, Discrete Optimization, 175 (2006), pp. 672–87.

[83] R. M. LUSBY AND J. LARSEN, *An exact method for the double TSP with multiple stacks*, Networks, 58 (2011), pp. 290–300.

[84] R. M. LUSBY, J. LARSEN, M. EHRGOTT, AND D. RYAN, *An exact method for the double TSP with multiple stacks*, International Transactions on Operations Research, 17 (2010), pp. 637–652.

[85] V. MAK AND A. ERNST, *New cutting-planes for the time- and/or precedence-constrained ATSP and directed VRP*, Mathematical Methods of Operations Research, 66 (2007), pp. 69–98.

[86] A. MALAPERT, C. GUERÉT, N. JUSSIEN, A. LANGEVIN, AND L.-M. ROUSSEAU, *Two-dimensional pickup and delivery routing problem with loading constraints*, in Proceedings of the First CPAIOR Workshop on Bin Packing and Placement Constraints (BPPC'08), Paris, France, 2008.

[87] H. MIN, *The multiple vehicle routing problem with simultaneous delivery and pick-up points*, Transportation Research Part A: Policy and Practice, 23 (1989), pp. 377–386.

[88] A. MINGOZZI, L. BIANCO, AND S. RICCIARDELLI, *Dynamic programming strategies for the traveling salesman problem with time window and precedence constraints*, Operations Research, 45 (1997), pp. 365–377.

[89] L. MINGYONG AND C. ERBAO, *An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows*, Engineering Applications of Artificial Intelligence, 23 (2010), pp. 188–195.

[90] S. MITROVIĆ-MINIĆ, R. KRISHNAMURTI, AND G. LAPORTE, *Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows*, Transportation Research Part B: Methodological, 38 (2004), pp. 669–685.

[91] S. MITROVIĆ-MINIĆ AND G. LAPORTE, *Waiting strategies for the dynamic pickup and delivery problem with time windows*, Transportation Research Part B: Methodological, 38 (2004), pp. 635–655.

[92] N. MLADENOVIĆ, D. UROŠEVIĆ, S. HANAFI, AND A. ILIĆ, *A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem*, European Journal of Operational Research, 220 (2012), pp. 270–285.

[93] F. A. T. MONTANÉ AND R. D. GALVÃO, *A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service*, Computers & Operations Research, 33 (2006), pp. 595–619.

[94] G. MOSHEIOV, *The travelling salesman problem with pick-up and delivery*, European Journal of Operational Research, 79 (1994), pp. 299–310.

[95] ——, *Vehicle routing with pick-up and delivery: Tour-partitioning heuristics*, Computers & Industrial Engineering, 34 (1998), pp. 669–684.

[96] G. NAGY AND S. SALHI, *Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries*, European Journal of Operational Research, 162 (2005), pp. 126–141.

[97] W. P. NANRY AND J. W. BARNES, *Solving the pickup and delivery problem with time windows using reactive tabu search*, Transportation Research Part B: Methodological, 34 (2000), pp. 107–121.

[98] M. NOWAK, O. ERGUN, AND C. C. WHITE, *Pickup and delivery with split loads*, Transportation Science, 42 (2008), pp. 32–43.

[99] I. OR, *Traveling Salesman-type Combinatorial Problems and Their Relation to the Logistics of Blood Banking*, PhD thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL, 1976.

[100] J. A. PACHECO, *Problemas de rutas con carga y descarga en sistemas LIFO: Soluciones exactas*, Estudios de Economía Aplicada, 3 (1995), pp. 69–86.

[101] ——, *Heuristico para los problemas de ruta con carga y descarga en sistemas LIFO*, SORT, Statistics and Operations Research Transactions, 21 (1997), pp. 153–175.

[102] G. PANKRATZ, *A grouping genetic algorithm for the pickup and delivery problem with time windows*, OR Spectrum, 27 (2005), pp. 21–41.

[103] S. PARRAGH, K. DOERNER, AND R. HARTL, *A survey on pickup and delivery problems part* I*: Transportation between customers and depot*, Journal für Betriebswirtschaft, 58 (2008), pp. 21–51.

[104] ——, *A survey on pickup and delivery problems part* II*: Transportation between pickup and delivery locations*, Journal für Betriebswirtschaft, 58 (2008), pp. 81–117.

[105] H. L. PETERSEN, C. ARCHETTI, AND M. G. SPERANZA, *Exact solutions to the double travelling salesman problem with multiple stacks*, Networks, 56 (2010), pp. 229–243.

[106] H. L. PETERSEN AND O. B. G. MADSEN, *The double travelling salesman problem with multiple stacks*, European Journal of Operational Research, 198 (2009), pp. 139–147.

[107] J. PRIVÉ, J. RENAUD, F. BOCTOR, AND G. LAPORTE, *Solving a vehicle-routing problem arising in soft-drink distribution*, Journal of the Operational Research Society, 57 (2006), pp. 1045–1052.

[108] H. N. PSARAFTIS, *A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem*, Transportation Science, 14 (1980), pp. 130–154.

[109] ——, *k-interchange procedures for local search in a precedence-constrained routing problem*, European Journal of Operational Research, 13 (1983), pp. 391–402.

[110] Y. QU AND J. F. BARD, *A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment*, Computers & Operations Research, 39 (2012), pp. 2439–2456.

[111] T. Raviv, M. Tzur, and I. Forma, *Static repositioning in a bike-sharing system: Models and solution approaches*, EURO Journal on Transportation and Logistics, 2 (2013), pp. 187–229.

[112] J. Renaud, F. F. Boctor, and G. Laporte, *Perturbation heuristics for the pickup and delivery traveling salesman problem*, Computers & Operations Research, 29 (2002), pp. 1129–1141.

[113] J. Renaud, F. F. Boctor, and J. Ouenniche, *A heuristic for the pickup and delivery traveling salesman problem*, Computers & Operations Research, 27 (2000), pp. 905–916.

[114] S. Ropke and J.-F. Cordeau, *Branch and cut and price for the pickup and delivery problem with time windows*, Transportation Science, 43 (2009), pp. 267–286.

[115] S. Ropke, J.-F. Cordeau, and G. Laporte, *Models and branch-and-cut algorithms for pickup and delivery problems with time windows*, Networks, 49 (2007), pp. 258–272.

[116] S. Ropke and D. Pisinger, *An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows*, Transportation Science, 40 (2006), pp. 455–472.

[117] ——, *A unified heuristic for a large class of vehicle routing problems with backhauls*, European Journal of Operational Research, 171 (2006), pp. 750–775.

[118] K. S. Ruland and E. Y. Rodin, *The pickup and delivery problem: Faces and branch-and-cut algorithm*, Computers & Mathematics with Applications, 33 (1997), pp. 1–13.

[119] D. Sáez, C. E. Cortés, and A. Núñez, *Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering*, Computers & Operations Research, 35 (2008), pp. 3412–3438.

[120] S. Salhi and G. Nagy, *A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling*, Journal of the Operational Research Society, 50 (1999), pp. 1034–1042.

[121] F. A. Santos, G. R. Mateus, and A. S. da Cunha, *The pickup and delivery problem with cross-docking*, Computers & Operations Research, 40 (2013), pp. 1085–1093.

[122] M. W. P. Savelsbergh, *An efficient implementation of local search algorithms for constrained routing problems*, European Journal of Operational Research, 47 (1990), pp. 75–85.

[123] M. W. P. Savelsbergh and M. Sol, *DRIVE: Dynamic routing of independent vehicles*, Operations Research, 46 (1998), pp. 474–490.

[124] P. Shaw, *Using constraint programming and local search methods to solve vehicle routing problems*, in Principles and Practice of Constraint Programming, M. Maher and J.-F. Puget, eds., Springer-Verlag, New York, 1998, pp. 417–431.

[125] X. Shi, F. Zhao, and Y. Gong, *Genetic algorithm for the one-commodity pickup-and-delivery vehicle routing problem*, in IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009, vol. 1, 2009, pp. 175–179.

[126] M. SIGURD, D. PISINGER, AND M. SIG, *Scheduling transportation of live animals to avoid the spread of diseases*, Transportation Science, 38 (2004), pp. 197–209.

[127] D. M. STEIN, *An asymptotic, probabilistic analysis of a routing problem*, Mathematics of Operations Research, 3 (1978), pp. 89–101.

[128] D. M. STEIN, *Scheduling dial-a-ride transportation systems*, Transportation Science, 12 (1978), pp. 232–249.

[129] A. SUBRAMANIAN, *Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems*, PhD thesis, Universidade Federal Fluminense, Rio de Janeiro, Brazil, 2012.

[130] A. SUBRAMANIAN AND M. BATTARRA, *An iterated local search algorithm for the travelling salesman problem with pickups and deliveries*, Journal of the Operational Research Society, 64 (2013), pp. 402–409.

[131] A. SUBRAMANIAN, L. M. A. DRUMMOND, C. BENTES, L. S. OCHI, AND R. FARIAS, *A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery*, Computers & Operations Research, 37 (2010), pp. 1899–1911.

[132] A. SUBRAMANIAN, E. UCHOA, AND L. S. OCHI, *A hybrid algorithm for a class of vehicle routing problems*, Computers & Operations Research, 40 (2013), pp. 2519–2531.

[133] A. SUBRAMANIAN, E. UCHOA, A. A. PESSOA, AND L. S. OCHI, *Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery*, Operations Research Letters, 39 (2011), pp. 338–341.

[134] A. SUBRAMANIAN, E. UCHOA, A. A. PESSOA, AND L. S. OCHI, *Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery*, Optimization Letters, 7 (2013), pp. 1569–1581.

[135] C. D. TARANTILIS, *Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking*, Optimization Letters, 7 (2013), pp. 1583–1596.

[136] C. D. TARANTILIS, A. K. ANAGNOSTOPOULOU, AND P. P. REPOUSSIS, *Adaptive path relinking for vehicle routing and scheduling problems with product returns*, Transportation Science, 47 (2013), pp. 356–379.

[137] S. TOULOUSE AND R. WOLFLER CALVO, *On the complexity of the multiple stack TSP, kSTSP*, Lecture Notes in Computer Science, Theory and Applications of Models of Computation, 5532 (2009), pp. 360–369.

[138] L. J. J. VAN DER BRUGGEN, J. K. LENSTRA, AND P. C. SCHUUR, *Variable-depth search for the single-vehicle pickup and delivery problem with time windows*, Transportation Science, 27 (1993), pp. 298–311.

[139] P. VENKATESHAN AND K. MATHUR, *An efficient column-generation-based algorithm for solving a pickup-and-delivery problem*, Computers & Operations Research, 38 (2011), pp. 1647–1655.

[140] T. VIDAL, T. G. CRAINIC, M. GENDREAU, AND C. PRINS, *A unified solution framework for multi-attribute vehicle routing problems*, European Journal of Operational Research, 234 (2014), pp. 658–673.

[141] A. C. WADE AND S. SALHI, *An investigation into a new class of vehicle routing problem with backhauls*, Omega, 30 (2002), pp. 479–487.

[142] ——, *An ant system algorithm for the mixed vehicle routing problem with backhauls*, in Metaheuristics, M. Resende, J. de Sousa, and A. Viana, eds., Kluwer Academic Publishers, Norwell, MA, 2004, pp. 699–719.

[143] N. A. WASSAN, G. NAGY, AND S. AHMADI, *A heuristic method for the vehicle routing problem with mixed deliveries and pickups*, Journal of Scheduling, 11 (2008), pp. 149–161.

[144] N. A. WASSAN, A. H. WASSAN, AND G. NAGY, *A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries*, Journal of Combinatorial Optimization, 15 (2008), pp. 368–386.

[145] M. WEN, J. LARSEN, J. CLAUSEN, J.-F. CORDEAU, AND G. LAPORTE, *Vehicle routing with cross-docking*, Journal of the Operational Research Society, 60 (2009), pp. 1708–1718.

[146] H. XU, Z.-L. CHEN, S. RAJAGOPAL, AND S. ARUNAPURAM, *Solving a practical pickup and delivery problem*, Transportation Science, 37 (2003), pp. 347–364.

[147] E. E. ZACHARIADIS AND C. T. KIRANOUDIS, *A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries*, Expert Systems with Applications, 38 (2011), pp. 2717–2726.

[148] E. E. ZACHARIADIS, C. D. TARANTILIS, AND C. T. KIRANOUDIS, *A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service*, Expert Systems with Applications, 36 (2009), pp. 1070–1081.

[149] ——, *An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries*, European Journal of Operational Research, 202 (2010), pp. 401–411.

[150] ——, *The pallet-packing vehicle routing problem*, Transportation Science, 46 (2012), pp. 341–358.

[151] F. ZHAO, S. LI, J. SUN, AND D. MEI, *Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem*, Computers & Industrial Engineering, 56 (2009), pp. 1642–1648.

[152] F.-G. ZHAO, J.-S. SUN, S.-J. LI, AND W.-M. LIU, *A hybrid genetic algorithm for the traveling salesman problem with pickup and delivery*, International Journal of Automation and Computing, 6 (2009), pp. 97–102.

[153] Y. ZHONG AND M. H. COLE, *A vehicle routing problem with backhauls and time windows: A guided local search solution*, Transportation Research Part E: Logistics and Transportation Review, 41 (2005), pp. 131–144.