9. VRP with Pickup and Delivery

Andreas Erdmann

The Vehicle Routing Problem

Chapter 9

VRP with Pickup and Delivery

Guy Desaulniers
Jacques Desrosiers
Andreas Erdmann
Marius M. Solomon
François Soumis

9.1 Introduction

In the *VRP with Pickup and Delivery* (VRPPD), a heterogeneous vehicle fleet based at multiple terminals must satisfy a set of transportation requests. Each request is defined by a pickup point, a corresponding delivery point, and a demand to be transported between these locations. The requested transport could involve goods or persons. This latter environment is called dial-a-ride. The objective function(s) generally minimizes system costs. The *VRPPD with Time Windows* (VRPPDTW) is a generalization of the VRPTW examined in Chapter 7. In the pickup (resp., delivery) version, the VRPTW is the particular case of the VRPPDTW where the destinations (resp., origins) are all at a common depot.

Problems in this class involve time constraints that establish time intervals during which service must take place at each stop, or that express user inconvenience and maximum ride time restrictions for passengers. For example, time windows for dial-a-ride problems model preferred pickup and delivery times specified by the customers. In addition to *time windows* to be satisfied at each stop, the VRPPD involves several other sets of constraints. These impose *visiting* each pickup and delivery stop exactly once, not exceeding the *capacity* of vehicles, and *coupling* the pickup and corresponding delivery stops on the same vehicle routes and impose visit *precedence* among each pickup stop and its associated drop-off stop. There are also *depot* constraints that ensure vehicles return to the appropriate terminals and *resource* restrictions on the number of drivers and vehicle types.

The VRPPDTW has a variety of practical applications, including the transport of the disabled and elderly, sealift and airlift of cargo and troops, and pickup and delivery for overnight carriers or urban services. Perspectives on this growing field were offered by Solomon and Desrosiers [52], Desrosiers et al. [12], and Savelsbergh and Sol [42]. Here, we extend their efforts by reviewing important recent developments and offering our view for future directions. We focus on static problems where all information is known with certainty when decisions are being made. We will also tangentially discuss certain dynamic VRPPDTWs where information becomes known during the planning horizon. A comprehensive examination of these latter problems is beyond the scope of this paper. Furthermore, two special cases of the VRPPDTW were addressed in Chapter 7. These are the transport of full loads, which can be appropriately modeled as a m-TSPTW, and backhaul environments where all deliveries precede pickups, which admits a VRPTW representation.

In section 9.2, we present a mathematical formulation of the VRPPDTW. In section 9.3, we overview the early work on the VRPPD and then examine route construction and improvement heuristics. We also address metaheuristics and neural network approaches. We conclude this section with a description of theoretical analyses derived for a few algorithms. Then, in section 9.4, we highlight optimization methods based on set-partitioning formulations or using dynamic programming. These are embedded in branch-and-bound algorithms or used as heuristic methods. We discuss additional applications in section 9.5, and put computational results with various approaches in perspective in section 9.6. Finally, we offer our concluding remarks in section 9.7.

9.2 Mathematical Formulation

We present below a request-based mathematical formulation for the VRPPDTW. It involves n requests with pickup and delivery stops as well as associated demands.

9.2.1 Construction of the Networks

Identify request i by two nodes, i and $n+i$, corresponding, respectively, to the pickup and delivery stops of the request. It is possible that different nodes may represent the same geographical location. Next, denote the set of pickup nodes by $P = \{1, \ldots, n\}$ and the set of delivery nodes by $D = \{n+1, \ldots, 2n\}$. Further, define $N = P \cup D$. If request i consists of transporting d_i units from i to $n+i$, let $\ell_i = d_i$ and $\ell_{n+i} = -d_i$.

Let K be the set of vehicles. Because not all vehicles can service all requests, each vehicle k has a specific set $N_k = P_k \cup D_k$ associated with it, where N_k, P_k, and D_k are appropriate subsets of N, P, and D, respectively. For each vehicle k, define now network $G_k = (V_k, A_k)$. Set $V_k = N_k \cup \{o(k), d(k)\}$ as the set of nodes inclusive of the origin, $o(k)$, and destination, $d(k)$, depots for vehicle k, respectively. The subset A_k of $V_k \times V_k$ comprises all feasible arcs. The capacity of vehicle k is given by C_k, and its travel time and cost between distinct nodes $i, j \in V_k$, by t_{ijk} and c_{ijk}, respectively.

Vehicle k is assumed to leave unloaded from its origin depot at time $a_{o(k)} = b_{o(k)}$. Each admissible pickup and delivery route for this vehicle corresponds to a feasible path from $o(k)$ to $d(k)$ in network G_k, visiting each node at most once. If the vehicle visits node $i \in N$, it must do so within the time window $[a_i, b_i]$ when the service time s_i must begin. Should it arrive too early, the vehicle is allowed to wait.

9.2.2 Formulation

The formulation involves three types of variable: binary flow variables x_{ijk}, equal to 1 if arc $(i, j) \in A_k$ is used by vehicle k, and 0 otherwise; time variables T_{ik} specifying when vehicle k starts the service at node $i \in V_k$; and variables L_{ik} giving the load of vehicle k after the service at node $i \in V_k$ has been completed. The formulation is as follows:

$$\min \sum_{k\in K} \sum_{(i,j)\in A_k} c_{ijk}x_{ijk} \tag{9.1}$$

subject to

$$\sum_{k\in K} \sum_{j\in N_k\cup\{d(k)\}} x_{ijk} = 1 \qquad \forall\, i \in P, \tag{9.2}$$

$$\sum_{j\in N_k} x_{ijk} - \sum_{j\in N_k} x_{j,n+i,k} = 0 \qquad \forall\, k \in K,\ i \in P_k, \tag{9.3}$$

$$\sum_{j\in P_k\cup\{d(k)\}} x_{o(k),j,k} = 1 \qquad \forall\, k \in K, \tag{9.4}$$

$$\sum_{i\in N_k\cup\{o(k)\}} x_{ijk} - \sum_{i\in N_k\cup\{d(k)\}} x_{jik} = 0 \qquad \forall\, k \in K,\ j \in N_k, \tag{9.5}$$

$$\sum_{i\in D_k\cup\{o(k)\}} x_{i,d(k),k} = 1 \qquad \forall\, k \in K, \tag{9.6}$$

$$x_{ijk}(T_{ik} + s_i + t_{ijk} - T_{jk}) \le 0 \qquad \forall\, k \in K,\ (i, j) \in A_k, \tag{9.7}$$

$$a_i \le T_{ik} \le b_i \qquad \forall\, k \in K,\ i \in V_k, \tag{9.8}$$

$$T_{ik} + t_{i,n+i,k} \le T_{n+i,k} \qquad \forall\, k \in K,\ i \in P_k, \tag{9.9}$$

$$x_{ijk}(L_{ik} + \ell_j - L_{jk}) = 0 \qquad \forall\, k \in K,\ (i, j) \in A_k, \tag{9.10}$$

$$\ell_i \le L_{ik} \le C_k \qquad \forall\, k \in K,\ i \in P_k, \tag{9.11}$$

$$0 \le L_{n+i,k} \le C_k - \ell_i \qquad \forall\, k \in K, n + i \in D_k, \tag{9.12}$$

$$L_{o(k),k} = 0 \qquad \forall\, k \in K, \tag{9.13}$$

$$x_{ijk} \ge 0 \qquad \forall\, k \in K,\ (i, j) \in A_k, \tag{9.14}$$

$$x_{ijk} \text{ binary} \qquad \forall\, k \in K,\ (i, j) \in A_k. \tag{9.15}$$

The linear objective function (9.1) minimizes the total travel cost. Constraints (9.2) and (9.3) impose that each request (i.e., the pickup and delivery nodes) is served exactly once and by the same vehicle. Constraints (9.4)–(9.6) characterize a multicommodity flow structure and ensure that each vehicle k starts from its origin depot $o(k)$ and terminates its route at its destination depot $d(k)$. Compatibility requirements between routes and schedules are handled by constraints (9.7), and (9.8) are the time window constraints. For each request, constraints (9.9) force the vehicle to visit the pickup node before the delivery node. Next, constraints (9.10) express the compatibility requirements between routes and vehicle loads, while (9.11)–(9.12) form the vehicle dependent capacity intervals at pickup and delivery nodes. Finally, the initial vehicle load is imposed by (9.13), and nonnegativity and binary requirements are given by (9.14) and (9.15), respectively. Constraint sets (9.3) through (9.15), as well as the objective function, are separable for each vehicle $k \in K$. This will be exploited later in the solution process based on mathematical decomposition.

This formulation limits route duration to at most $b_{d(k)} - a_{o(k)}$. Note also that constraints (9.7), along with the time window constraints, allow a vehicle to wait before its visit to a node. There is no penalty on waiting time, and the arrival time at node j can be calculated as

$$x_{ijk} = 1 \Rightarrow T_{jk} = \max\{a_j,\ T_{ik} + s_i + t_{ijk}\}, \qquad (i,\ j) \in A_k.$$

The minimization of the fleet size also can be considered in this formulation by incorporating a large cost in the values $c_{o(k),jk}$ for $j \in P_k$. In this case, one should include arc $(o(k), d(k))$ in A_k at zero cost to allow for a vehicle not to be used. When the fleet is heterogeneous, cost coefficients can be assigned unequal weights to encourage the use of certain classes of vehicles over others.

As proposed by Dumas, Desrosiers, and Soumis [17] (see also Desaulniers et al. [11] for a general discussion), the above linear objective function can easily be replaced by a more general nonlinear function. For example, let $c_k(L_{ik}) > 0$, $i \in V_k$, denote a nondecreasing function of the total load transported on vehicle k, just after the service is completed at node i. This function acts as a penalty factor on the travel cost and (9.1) can be replaced by

$$\min \sum_{k \in K} \sum_{(i,j) \in A_k} c_k(L_{ik}) c_{ijk} x_{ijk}.$$

9.2.3 Service Quality

For goods transportation, it is sufficient that the objective function account for the number and type of vehicles and routing costs as described above. However, for the transport of persons, another essential component is the quality of the service provided by the system to its users. Researchers have addressed these three objectives either sequentially—beginning with fleet size minimization and ending with service quality maximization—or in parallel. Service quality commonly has been measured by user inconvenience, which models discrepancies between times requested by customers and actual pickups and deliveries. Constraints (9.8) implicitly define the service quality level. Quality increases with reductions in time window width. Yet, total service quality—zero user inconvenience—hardly can be expected given that peak demand generally leads to unrealistic capacity requirements. Nevertheless, high quality levels can be achieved by relaxing constraints (9.8) and penalizing them in the objective function, or explicitly optimizing pickup and delivery times once vehicle schedules have been obtained. Fixed-route optimization was discussed at length in Dumas, Soumis, and Desrosiers [18] and Desrosiers et al. [12].

9.2.4 Reduction of the Network Size

Reduction of the network size is a preliminary phase to solving the VRPPDTW by either a heuristic or an optimization-based approach. The two main steps are to narrow the widths of time windows and to eliminate the inadmissible arcs. The latter comprises a variety of techniques based on restrictions imposed by precedence, vehicle capacity, time windows, and identical location (if the travel costs satisfy the triangle inequality). Details are provided by Dumas, Desrosiers, and Soumis [17].

9.3 Heuristics

The early work on VRPPD was conducted for dial-a-ride scenarios. It was first examined by Wilson et al. [59], Wilson and Weissberg [60], and Wilson and Colvin [58] and was motivated by the demand-responsive transportation systems of Haddonfield, NJ, and Rochester, NY. This stream of work introduced the fundamental concepts of building tours through sequential insertion of customers and the general form of the objective function.

9.3.1 Construction and Improvement

A parallel insertion heuristic similar to that of Wilson et al. was proposed by Roy et al. [38, 39] for the multiple VRPPDTW in the context of the transportation of disabled persons. Since a fair amount of requests are known in advance, these are used by means of time-spatial proximity criteria to create initial routes for all vehicles starting at the beginning of the day. New requests are inserted in the set of existing routes or new routes initialized as needed.

Jaw et al. [25] explicitly considered time windows. Each customer specifies either a desired pickup time or a delivery time. The system operates under three types of service quality constraints: if a customer has specified a pickup (delivery) time, the actual pickup (delivery) should not take place earlier (later) than the desired time, and the waiting time, as well as the ride time, should not exceed a given maximum compared to the direct ride time. In addition, a vehicle is not allowed to be idle when carrying passengers. The authors developed an insertion heuristic where customers demanding service are selected in order of increasing earliest pickup time and inserted in the vehicle schedule with the lowest additional costs, taking into account inconvenience to the new customer and operational costs. Schedule blocks, that is, continuous periods of active vehicle time between two successive periods of vehicle slack time, play an important role in finding feasible insertions, since they allow the examination of all possible schedule sequences. Psaraftis [30] reported on a comparison with a previous approach developed by the same authors.

Madsen, Ravn, and Rygaard [29] implemented a generalized version of this approach for a partly dynamic dial-a-ride problem. Their algorithm can minimize vehicle waiting time as well as introduce breaks. Using more detailed job properties instead of schedule blocks, the number of insertions to examine was reduced. Requests known in advance are considered static, while real-time requests are handled sequentially.

Local search for the VRPPD was first considered by Psaraftis [34], who extended the ideas of Lin [27] and Lin and Kernighan [28]. A decade later, Van der Bruggen, Lenstra, and Schuur [57] presented another local search heuristic for the 1-VRPPDTW (single vehicle) for minimizing route duration. Their approach is based on a variable-depth search similar to the technique of Lin and Kernighan [28] for the TSP and simulated annealing. The algorithm involves two phases, both using arc exchange procedures. In the construction phase, it tries to find an initial feasible route allowing infeasibility and penalizing the violation of restrictions in the objective function. In the improvement phase, the method considers solely feasible solutions and tries to minimize route length.

9.3.2 Clustering Algorithms

Clustering algorithms use customer proximity to guide and possibly simplify the routing aspect. Geographical closeness among customers is used either a priori or in parallel with the routing process to cluster them. An early approach was that of Cullen, Jarvis, and Ratliff [8], who proposed an interactive optimization approach for the multiple vehicle dial-a-ride problem where customers are serviced by a homogeneous fleet. For the same context, Bodin and Sexton [4] developed a traditional *cluster-first, route-second* approach. Single vehicle cases are solved using the method of Sexton and Bodin [44, 45].

Since each customer represents a set of two locations (pickup and delivery), it is difficult to generate high-quality clusters without incorporating some routing information. Hence, Dumas, Desrosiers, and Soumis [16] suggested the use of miniclusters, that is, customers that can form an appealing route segment. Their sequential approach is revisited in Desrosiers et al. [14], where insertions are performed in parallel. Further work in this direction was performed by Ioachim et al. [24], who used mathematical optimization techniques to globally define a set of miniclusters. These were generated by solving an m-VRPPDTW with an enhanced version of the algorithm of Dumas, Desrosiers, and Soumis [17].

9.3.3 Metaheuristics

In contrast to the generic VRP and its variants with time windows or one-sided requests, where high quality solutions were obtained by using metaheuristics (see Chapter 7), the literature is scant for the VRPPD.

Gendreau et al. [22] suggested a dynamic pickup and delivery m-TSP with soft time windows to model courier services for the same-day local pickup and delivery of small items. The objective function to be minimized is a weighted sum of the total travel time, lateness, and overtime. To solve this problem, the authors proposed an adaptive memory-based tabu search. They used a neighborhood structure based on the concept of ejection chains: a request (i.e., the pickup and delivery location) is chosen, taken from its route (ejected), and moved to another route (inserted), where another request is forced to move to another route. The problem of finding the best chain or cycle of ejection or insertion moves over the current set of routes is modeled as a constrained shortest-path problem and solved heuristically. To intensify the search, the starting solution is decomposed into disjoint subsets of adjacent routes, each to be processed by a different tabu search. A two-level parallelization is proposed where the master manages the adaptive memory and produces solutions from it. These solutions are transmitted to slave processes that improve them by performing tabu search and return the best solution to the master.

Toth and Vigo [56] described a procedure for transporting disabled persons in an urban area using a mixed fleet, an instance of the dial-a-ride problem. They used an objective function that encompasses fixed vehicle costs, routing costs, and user inconvenience penalties. That is, they discouraged the use of the more costly vehicle type—taxis—and also relaxed the time window constraints in the objective function using piecewise linear user inconvenience penalties. The algorithm first estimates the minimum number of routes necessary to undertake a given percentage of the trips based solely on capacity considerations. Each route is then initialized with a trip that maximizes a certain score. Next, unrouted trips are inserted into routes or new routes are created, if needed, by solving a minimum-cost rectangular assignment problem on the insertion cost matrix. This is composed of elements representing the additional cost of inserting a certain trip in a given route in the best feasible position.

The heuristic is then improved using Tabu Thresholding, which involves alternating between a phase where a local optimum is reached and another where moves away from it are considered. Both phases rely on an iterative candidate list method that selects a subset of moves at each iteration from a family of subsets partitioning the search neighborhood of the current solution. The subsets of trip movements and exchanges that form a partition are generated according to Toth and Vigo [55].

9.3.4 Neural Network Heuristics

A different approach was proposed by Shen et al. [47]. It is an expert consulting system for a dispatcher working in a courier service. It consists of two modules: dispatching and learning. The former assists the dispatcher in allocating each new request to one of the available drivers. It estimates distances and travel times and helps evaluate the consequences of insertions. The latter suggests "good" drivers to service the new requests. This module is based on a backpropagation neural network. The network is trained using decision data from a dispatcher, while its performance is evaluated by comparison to the dispatcher's decisions.

9.3.5 Theoretical Analysis of Algorithms

A stream of research conducted by Daganzo [9] and several coauthors attempted to obtain analytical insight into distribution patterns at the system design level. That is, they tried to characterize broad routing strategies independent of specific customer locations but rather in terms of problem characteristics. This kind of analysis could provide guidelines for districting and cost estimation.

The intrinsic complexity of this problem class made it very difficult to analyze specific solution methodologies beyond empirical testing. Stein [53] was the first to present a probabilistic analysis of a simple algorithm. This constructs a VRPPD tour by concatenating two TSP tours, one through the origins and the other through the destinations. He proved that this algorithm is asymptotically bounded by 1.06. Later, Psaraftis [32] showed that an adaptation of the minimum spanning tree heuristic for the TSP has a worst-case performance ratio of 3.0. He further showed that its practical performance is better than that of Stein's [53] method.

Gendreau, Laporte, and Vigo [23] analyzed the capacitated 1-VRPPD (or TSPPD) on trees and cycles. They show that these problems can be solved optimally in linear time. For the TSPPD on a tree, the proposed $O(n)$ exact algorithm traverses the tree depth-first and visits all subtrees involving larger deliveries than pickups before visiting those that require positive net pickups. Within a subtree, vertices are visited in the net-deliveries-first order as well. When the underlying network is a cycle, the authors proved that the optimal solution is the better of the following two solutions. The first is obtained by eliminating the largest cost arc and traversing the rest twice, and the second involves traversing all arcs either one or three times. Note that in the former solution, once an arc has been eliminated, the cycle becomes a line and therefore a special case of a tree.

The algorithms proposed for these special cases are then turned into approximate methods for the general problem. Specifically, the tree-based heuristic applies the above procedure on a spanning tree for the underlying network. The cycle-based heuristic first generates a Hamiltonian tour by solving the TSP relaxation. Then, the above exact algorithm is used to solve the 1-VRPPD on this cycle. The solution obtained is an Eulerian tour that is transformed in a Hamiltonian solution by means of shortcuts. The authors then proved that on undirected, complete, and triangular networks, the tree- (cycle-) based heuristic has a tight worst-case performance ratio of 2 (3).

9.4 Optimization-Based Approaches

In this section we describe several scenarios, since different authors have analyzed slightly different VRPPD variants.

9.4.1 Single Vehicle Cases

Scenarios involving a single vehicle have received attention due to their intrinsic importance and to gain insight into situations where multiple vehicles are used.

9.4.1.1 Benders' Decomposition

Sexton and Bodin [44, 45] proposed a robust heuristic based on Benders' decomposition for the single vehicle VRPPD with one-sided time windows, i.e., each request has only a desired delivery time. They used an objective function that minimizes total customer inconvenience in a manner similar to Psaraftis [31, 33], whose work is described below. The problem decomposes into a routing problem (the Benders' master) and a scheduling subproblem. While the scheduling component is a network flow problem that can be solved very efficiently, integer solutions for the routing problem are derived using a route improvement procedure. An extension to the soft time window case was given in Sexton and Choi [46]. The objective function was modified such that missed time windows incur penalties.

9.4.1.2 Dynamic Programming

An exact backward dynamic programming algorithm for solving the single vehicle dial-a-ride problem was suggested by Psaraftis [31]. Customers request service by phone and are served according to their calling order. The objective function is a weighted combination of the total route length and the total customer inconvenience (given as the sum of waiting and riding times). The algorithm was later modified by Psaraftis [33] to a forward dynamic programming approach for the variant with time windows. The time complexity for both algorithms is $O(n^2 3^n)$. For the TSP with precedence constraints, Bianco et al. [3] were able to develop an improved backward dynamic program by using a lower bound to prune the search space.

An exact forward dynamic programming approach to minimize the total distance traveled in the static single vehicle VRPPD with time windows was developed by Desrosiers, Dumas, and Soumis [13]. The mathematical formulation can easily be derived from (9.1)–(9.15) by letting $|K|$ equal to 1. The authors capitalize on the fact that all nodes must be visited by only one vehicle. A state (S, i) is defined if there exists a feasible path that starts at the depot node o, visits all the nodes in $S \subseteq N$, and ends at node $i \in S$. A path is feasible if it satisfies vehicle capacity, precedence, and time window constraints. For each state (S, i), two-dimensional labels are defined for each path from o to i, with time and distance components (the load can be retrieved from the visited nodes of S). As usual, only Pareto-optimal labels are kept. Such a label of state (S, i) is then tested to check if there is a feasible path starting at i and visiting all the remaining nodes. From these unvisited nodes, if there is one that cannot be visited, the label is eliminated. Several efficient criteria were proposed by the authors, such as testing only for the unvisited node with the earliest start of service.

9.4.1.3 Polyhedral Approach

The only polyhedral approach for the VRPPD is given by Ruland [40] and Ruland and Rodin [41]. It is based on the solution of the TSP with precedence constraints. The formulation for the single-vehicle VRPPD without capacity constraints is defined on an undirected graph $G = (V, E)$ and uses only the binary flow variables x_e, $e \in E$, which are equal to 1 if the arc e is used and 0 otherwise.

$$\min \sum_{e \in E} c_e x_e \tag{9.16}$$

subject to

$$x_{od} = 1, \tag{9.17}$$

$$x(\delta(\{v\})) = 2 \qquad \forall\, v \in V, \tag{9.18}$$

$$x(\delta(U)) \geq 2 \qquad \forall\, U \in \mathcal{U}, \tag{9.19}$$

$$x(\delta(U)) \geq 4 \qquad \forall\, U \in \mathcal{U}'_p, \tag{9.20}$$

$$0 \leq x_e \leq 1 \qquad \forall\, e \in E, \tag{9.21}$$

$$x \in \mathbf{Z}^{|E|}, \tag{9.22}$$

where $\mathcal{U} = \{U \subset V : o \in U\}$ is the set of all vertex sets containing the vehicle origin depot, and $\mathcal{U}'_p = \{U \subset \mathcal{U} : d \notin U, \exists i \in P, i \notin U, n+i \in U\}$.

The objective function (9.16) minimizes the total travel cost. Constraint (9.17) gives a reference direction. Removing this arc from a feasible solution gives the vehicle route. The degree equality constraints (9.18) require the degree of each vertex $v \in V$ in a feasible solution to equal 2, and the subtour elimination constraints (9.19) forces biconnectedness. Constraints (9.20) are the precedence constraints defined on cut sets: whenever the origin depot and a delivery location $n + i \in D$ are in a cut set, and the corresponding destination depot and pickup location $i \in P$ are not, respectively, the flow crossing this cut set must be greater than or equal to 4. Finally, binary flow requirements are given by (9.21) and (9.22). Several enhancements to this formulation are proposed by Ruland and Rodin.

9.4.2 Multiple Vehicle Cases

To model multiple vehicle cases, researchers have relied on the set-partitioning model. The scale of the models generated has in turn naturally lead to decomposition approaches based on column generation. Before presenting them, we begin with a different solution method that reduces the complexity of the set-partitioning model through additional simplifying assumptions.

9.4.2.1 Matching-Based Approach

Derigs and Metz [10] dealt with a special case of the VRPPD where only one-sided time windows are present and all customers play the role of delivery and pickup points. During the first phase of the planning period, the customers receive deliveries from the depot, while pickup and transportation of goods to the depot takes place in the second phase. The problem is formulated as a (high-dimensional) set-partitioning problem with two additional nontrivial sets of side constraints. Under the assumption that the number of customers that can be supplied by a single vehicle in both the delivery and the pickup phases is at most two, the problem is reduced to a matching problem with side constraints. The problem is still NP-complete, but good approximate solutions can be constructed in acceptable computational time by relaxation and the application of optimization techniques from nonsmooth optimization and matching algorithms.

9.4.2.2 Column Generation

Dumas, Desrosiers, and Soumis [17] presented an exact algorithm for the VRPPDTW. They used a set-partitioning model and proposed a column-generation approach with a constrained shortest-path subproblem to build admissible routes. This model can easily be derived from (9.1)–(9.15) by applying the appropriate Dantzig–Wolfe decomposition, i.e., keeping (9.1) and (9.2) in the master problem and using the remaining constraint sets to define the constrained shortest path subproblem.

To present their formulation, assume a heterogeneous fleet of vehicles. For vehicle $k \in K$, let Ω_k be the set of feasible pickup and delivery routes and c_{rk} the cost of route r. The binary constants a_{irk} are equal to 1 if route r of vehicle k includes request i and 0 otherwise. The formulation uses just one type of variable: the binary variable y_{rk} equals to 1 if route r is used for vehicle k and 0 otherwise. The VRPPDTW can now be stated as follows:

$$\min \sum_{k \in K} \sum_{r \in \Omega_k} c_{rk} y_{rk} \tag{9.23}$$

subject to

$$\sum_{k \in K} \sum_{r \in \Omega_k} a_{irk} y_{rk} = 1 \qquad \forall\, i \in P, \tag{9.24}$$

$$\sum_{r \in \Omega_k} y_{rk} = 1 \qquad \forall\, k \in K, \tag{9.25}$$

$$y_{rk} \text{ binary} \qquad \forall\, k \in K,\ r \in \Omega_k. \tag{9.26}$$

The reader can observe that the covering constraints (9.24) are imposed only on the pickup nodes since the pairing of pickup and delivery nodes within the same route also ensures the service at the delivery nodes. Similar to the column-generation algorithm described for the VRPTW in Chapter 7, new routes are generated by transferring the current dual variables of constraints (9.24) and (9.25) to the vehicle networks G_k and using a specialized dynamic programming algorithm for the pickup and delivery shortest path subproblems. This algorithm (see Dumas, Desrosiers, and Soumis [17]) uses three-dimensional labels and allows for multiple visits at the same node. Indeed, the smallest request cycle must satisfy a sequence of at least five nodes, $i \to n+i \to j(\neq i) \to i \to n+i$, as arc $(n+i, i)$ does not exist. This is unlikely in practice, when time windows are small compared to travel times. The optimal solution to the linear relaxation of the master problem is reached when routes with negative marginal cost can no longer be found.

To obtain integer solutions, a branch-and-bound tree is used. To avoid generating a huge enumeration tree when setting to 0 or 1 variables x_{ijk} from the multicommodity flow formulation stated in section 9.2, Dumas, Desrosiers, and Soumis propose another branching strategy that can be applied directly to the requests. Order variables O_{ij}, for $i, j \in P$, are introduced to allow to branch on pickup sequences. When branching on these, the information can be inserted in the master problem by deleting variables associated with forbidden paths and in the subproblem by using a fourth label dimension. This new dimension represents the last pickup node visited, and order constraints can be easily checked when extending a label. Very good solutions were obtained by exploring only a few nodes, typically fewer than 10, of the branch-and-bound tree.

Savelsbergh and Sol [43] and Sol [48] used a similar approach that differs in the aspects below. The pricing subproblem is solved by heuristic construction and insertion algorithms using the cheapest insertion cost, in addition to a dynamic programming approach. Further, branching is done on binary variables x_{ik} for $i \in P_k$ and $k \in K$,

$$x_{ik} = \sum_{r \in \Omega_k} a_{irk} y_{ik},$$

indicating which fraction of request i is served by vehicle k. Also, the current fractional solution is used for a primal heuristic: if the value x_{ik} is large, it is likely that transportation request i will be served by vehicle k. Using the sequence of decreasing x_{ik}, routes are created from scratch using the insertion algorithm and are, if a feasible solution is found, subjected to improvement heuristics.

The above methodology was extended to a combined inventory management and VRPPDTW in a sealift environment by Christiansen and Nygreen [6] and [7]. The problem involves transporting a bulk product by ship from production to consumption harbors. The amounts to be transported depend on production rates, inventory levels, and ship-harbor compatibility. Using variable splitting (see Chapter 7), the authors were able to separate the problem by ship and production harbor. The former paper discusses the overall solution approach, while the latter focuses on the ship and inventory subproblems.

An essential characteristic of the above optimization algorithms based on column generation is their flexibility. That is, they are primal methods that provide feasible solutions early in the process and hence can easily be used as approximation methods by early termination. In conjunction with sophisticated column-generation management and the use of heuristics within the overall optimization scheme whenever appropriate, these algorithms can solve realistic-size problems. Another key feature of such methods is that they are readily amenable to reoptimization. This makes them viable approaches for dynamic environments.

9.5 Applications

The majority of applications have occurred in sealift and airlift environments and the transportation of elderly or disabled in urban areas. Additional applications have been considered in school bus routing and scheduling. This special VRPPDTW case is discussed in Chapter 7. In early work in the sealift context, Psaraftis et al. [35] suggested a heuristic based on sequentially solving a transportation problem for each time slice of the planning horizon to decrease total tardiness. Its solutions can be improved using the method described by Thompson and Psaraftis [54]. Fisher and Rosenwein [21] examined the pickup and delivery of bulk cargoes. They presented an interactive optimization system based on the branch-and-bound solution of a set packing problem (see Fisher and Kedia [20]). The method extends prior work on a truck-scheduling problem described by Bell et al. [2] and Fisher et al. [19] to this context. More recently, Christiansen [5] considered a ship-planning application for the sealift environment described in the previous section. The author solved the problem by Danzig–Wolfe decomposition embedded in an overall branch-and-bound scheme.

For airlift scenarios, Rappoport et al. [36] and [37] proposed an airlift planning algorithm that assigns payload to aircraft during a long planning horizon. This solution can then be used to initialize algorithms that provide more detailed assignments and schedules for shorter horizons. Such an approach was suggested by Solanki and Southworth [49], who enhanced Solomon's [50] insertion heuristic to modify an existing military airlift schedule. Solomon's method has also been extended to a large-scale larvicide control program by Solomon et al. [51]. In this multiperiod 1-VRPPDTW, helicopters must discharge several types of larvicide in rivers to fight larvae growth. To solve the problem, the authors created miniclusters by larvicide type and then applied the insertion heuristic.

Another application area is the scheduling of vehicles for transportation of elderly or disabled. In an early application, Alfa [1] used [25] for this purpose. Later approaches include those of Ioachim et al. [24] and Toth and Vigo [56]. The former authors showed the benefits of their method using data from the city of Toronto. The latter produced very good results in service quality and overall cost compared to the manual schedules used in the city of Bologna. Recently, Savelsbergh and Sol [43] reported encouraging results on data from a road transportation environment in Europe.

9.6 Computational Results

The computational experience reported on the VRPPD indicates that algorithms capable of solving larger or more difficult problems are constantly being proposed. The papers by Solomon and Desrosiers [52], Desrosiers et al. [12], and Savelsbergh and Sol [42] illustrate this trend by discussing the computational capabilities specific to the different methods available up to publication time. Nevertheless, the relative evaluation of competing approaches is much more difficult in this environment. This is because a benchmark problem set has not been developed for the VRPPD as it has for the generic VRP or VRPTW, for example. The primary reason is the multitude of problem variants that the literature has addressed. Generally, much of the work has stemmed from applications that induced modeling differences. For example, the manner in which service quality is represented in the objective function or the constraints is often situation specific. Therefore, researchers have preferred to test their methods on data simulated from the real-world setting they were analyzing. In addition, they used the actual data to compare their methods to manual solutions in use.

The intricacy of this problem class has hampered efforts to optimally solve problems with more than tens of requests. The methodology proposed by Ruland and Rodin [41] for the 1-VRPPD was able to solve problems involving 15 requests, while that of Desrosiers, Dumas, and Soumis [13] solved 1-VRPPDTW instances with 40 requests. In the multiple-vehicle case with time windows, the algorithms of Dumas, Desrosiers, and Soumis [17] and Savelsbergh and Sol [43] have been successful on problems involving about 50 requests.

Heuristic approaches have been effective in solving larger size problems found in practice. Generally, approximate methods are able to solve problems with hundreds of requests. In particular, Toth and Vigo [56] have shown their approach to be computationally viable for a problem consisting of more than 300 requests. Larger scale instances have been tackled by decomposing the original problems geographically, using miniclusters, or temporally, using time slices. The resulting problems consist of hundreds of requests. Jaw et al. [25] reported solving a real dial-a-ride problem with more than 2600 requests and 20 vehicles, while Ioachim et al. [24] handled more than 2500 requests. The method proposed by Dumas, Desrosiers, and Soumis [16] successfully solved problems with more than 3500 requests.

9.7 Conclusions

In this chapter we described the research conducted on the VRPPD over the last 20 years. Not surprising, the development of the field and the level of methodological sophistication has paralleled that of other routing variants. As practical instances of the VRPPD are large scale, researchers have favored heuristic approaches. In particular, various insertion and local search improvement procedures frequently have been proposed. While more intricate methods, such as metaheuristics, have been developed over time, these have not yielded the same benefits as other VRP variants. Parallel computing may be the answer to making them computationally viable. In addition, the work of Toth and Vigo [56] shows promise for composite heuristics that embed tabu search within an insertion or improvement solution framework. We expect interest in such methods to intensify. Future research could also benefit from the generation of a realistic benchmark problem set on which competing approaches could be evaluated.

Research on exact algorithms has attempted to exploit special problem structure and the progress in computing technology in a manner similar to that used in other VRP sectors. Yet, because of the ancillary complexity of the VRPPD, much work remains to be done. Recent ideas used successfully elsewhere, such as valid inequalities (Kohl et al. [26]) and master problem acceleration strategies using bounded perturbation variables (du Merle et al. [15]), could provide the impetus for future advances. (see Chapter 7 for a discussion and additional details). More extensive research on polyhedral approaches could also provide valuable insight in this direction.

Recent advances in the telecommunications and information infrastructure have generated noteworthy interest in dynamic aspects of the VRPPD. For example, vehicle diversion is becoming common practice since satellite systems can provide real-time information. This, coupled with current business emphasis on responsiveness and cost reduction, suggests that interest in this area will only magnify. These developments open exciting new research arenas in this field. We hope this chapter provided its readers with a starting basis for their research on the challenging problems ahead.

Bibliography

[1] A.S. Alfa. Scheduling of vehicles for transportation of elderly. *Transportation Planning and Technology*, 11:203–212, 1986.

[2] W. Bell, L. Dalberto, M.L. Fisher, A. Greenfield, R. Jaikumar, P. Kedia, R. Mack, and P. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13:4–23, 1983.

[3] 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*, 32:19–31, 1994.

[4] L.D. Bodin and T. Sexton. The multi-vehicle subscriber dial-a-ride problem. *TIMS Studies in the Management Sciences*, 22:73–86, 1986.

[5] M. Christiansen. Decomposition of a combined inventory routing and time constrained ship routing problem. *Transportation Science*, 33:3–16, 1999.

[6] M. Christiansen and B. Nygreen. A method for solving ship routing problems with inventory constraints. *Annals of Operations Research*, 81:357–378, 1998.

[7] M. Christiansen and B. Nygreen. Modelling path flows for a combined routing and inventory management problem. *Annals of Operations Research*, 82:391–412, 1998.

[8] F. Cullen, J. Jarvis, and D. Ratliff. Set partitioning based heuristics for interactive routing. *Networks*, 11:125–144, 1981.

[9] C. Daganzo. *Logistics systems analysis*. Springer-Verlag, Heidelberg, Germany, 1991.

[10] U. Derigs and A. Metz. A matching-based approach for solving a delivery/pick-up VRP with time constraints. *OR-Spektrum*, 14:91–106, 1992.

[11] G. Desaulniers, J. Desrosiers, I. Ioachim, M.M. Solomon, F. Soumis, and D. Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T.G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, Kluwer, Boston, MA, 1998, pp. 57–93.

[12] J. Desrosiers, Y. Dumas, M.M. Solomon, and F. Soumis. Time constrained routing and scheduling. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Routing*, *Handbooks in Operations Research and Management Science* 8, North-Holland, Amsterdam, 1995, pp. 35–139.

[13] 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*, 6:301–325, 1986.

[14] J. Desrosiers, Y. Dumas, F. Soumis, S. Taillefer, and D. Villeneuve. An algorithm for mini-clustering in handicapped transport. Technical Report Cahiers du GERAD G–91–02, École des Hautes Études Commerciales, Montréal, Canada, 1991.

[15] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194:229–237, 1999.

[16] Y. Dumas, J. Desrosiers, and F. Soumis. Large scale multi-vehicle dial-a-ride problems. Technical Report Cahiers du GERAD G–89–30, École des Hautes Études Commerciales, Montréal, Canada, 1989.

[17] Y. Dumas, J. Desrosiers, and F. Soumis. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54:7–22, 1991.

[18] Y. Dumas, F. Soumis, and J. Desrosiers. Optimizing the schedule for a fixed vehicle path with convex inconvenience costs. *Transportation Science*, 24:145–152, 1990.

[19] M.L. Fisher, A. Greenfield, R. Jaikumar, and J. Lester. A computerized vehicle routing application. *Interfaces*, 12:42–52, 1982.

[20] M.L. Fisher and P. Kedia. Optimal solution of set covering/partitioning problems using dual heuristics. *Management Science*, 36:674–688, 1990.

[21] M.L. Fisher and M.B. Rosenwein. An interactive optimization system for bulk-cargo ship scheduling. *Naval Research Logistic Quarterly*, 35:27–42, 1989.

[22] M. Gendreau, F. Guertin, J.-Y. Potvin, and R. Séguin. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Technical Report CRT–98–10, Centre de recherche sur les transports, Université de Montréal, Canada, 1998.

[23] M. Gendreau, G. Laporte, and D. Vigo. Heuristics for the traveling salesman problem with pickup and delivery. *Computers and Operations Research*, 26:699–714, 1999.

[24] I. Ioachim, J. Desrosiers, Y. Dumas, M.M. Solomon, and D. Villeneuve. A request clustering algorithm for door-to-door handicapped transportation. *Transportation Science*, 29:63–78, 1995.

[25] J. Jaw, A. Odoni, H. Psaraftis, and N. Wilson. A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with windows. *Transportation Research B*, 20:243–257, 1986.

[26] N. Kohl, J. Desrosiers, O.B.G. Madsen, M.M. Solomon, and F. Soumis. k-Path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33:101–117, 1999.

[27] S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44:2245–2269, 1965.

[28] S. Lin and B.W. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498–516, 1973.

[29] O.B.G. Madsen, H.F. Ravn, and J.M. Rygaard. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, 60:193–208, 1995.

[30] H. Psaraftis. Scheduling large-scale advance-request dial-a-ride systems. *American Journal of Mathematical and Management Sciences*, 6:327–367, 1986.

[31] H.N. Psaraftis. A dynamic programming solution to the single-vehicle, many-to-many, immediate request dial-a-ride problem. *Transportation Science*, 14:130–154, 1980.

[32] H.N. Psaraftis. Analysis of an $o(n^2)$ heuristic for the single vehicle many-to-many euclidean dial-a-ride problem. *Transportation Research B*, 17:133–145, 1983.

[33] H.N. Psaraftis. An exact algorithm for the single-vehicle, many-to-many dial-a-ride problem with time windows. *Transportation Science*, 17:351–357, 1983.

[34] H.N. Psaraftis. k-Interchange procedures for local search in a precedence-constrained routing problem. *European Journal of Operational Research*, 13:391–402, 1983.

[35] H.N. Psaraftis, J.B. Orlin, D. Bienstock, and P.M. Thompson. Analysis and solution algorithms of sealift routing and scheduling problems: Final report. Technical Report 1700-85, MIT, Sloan School of Management, Cambridge, MA, 1985.

[36] H.K. Rappoport, L.S. Levy, B.L. Golden, and K. Toussaint. A planning heuristic for military airlift. *Interfaces*, 22:73–87, 1992.

[37] H.K. Rappoport, L.S. Levy, K. Toussaint, and B.L. Golden. A transportation problem formulation for the MAC airlift planning problem. *Annals of Operations Research*, 50:505–523, 1994.

[38] S. Roy, J.-M. Rousseau, G. Lapalme, and J.A. Ferland. Routing and scheduling for the transportation of disabled persons—the algorithm. Technical Report TP 5596E, Centre de Recherche sur les Transports, Montréal, Canada, 1984.

[39] S. Roy, J.-M. Rousseau, G. Lapalme, and J.A. Ferland. Routing and scheduling for the transportation of disabled persons—the tests. Technical Report TP 5598E, Centre de Recherche sur les Transports, Montréal, Canada, 1984.

[40] K.S. Ruland. Polyhedral solution to the pickup and delivery problem. Ph.D. thesis, Washington University, St. Louis, MO, 1995.

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

[42] M.W.P. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29:17–29, 1995.

[43] M.W.P. Savelsbergh and M. Sol. Drive: Dynamic routing of independent vehicles. *Operations Research*, 46:474–490, 1998.

[44] T.R. Sexton and L.D. Bodin. Optimizing single vehicle many-to-many operations with desired delivery times. I: Scheduling. *Transportation Science*, 19:378–410, 1985.

[45] T.R. Sexton and L.D. Bodin. Optimizing single vehicle many-to-many operations with desired delivery times. II: Routing. *Transportation Science*, 19:411–435, 1985.

[46] T.R. Sexton and Y.-M. Choi. Pickup and delivery of partial loads with "soft" time windows. *American Journal of Mathematical and Management Sciences*, 6:369–398, 1986.

[47] Y. Shen, J.-Y. Potvin, J.-M. Rousseau, and S. Roy. A computer assistant for vehicle dispatching with learning capabilities. *Annals of Operations Research*, 61:189–211, 1995.

[48] M. Sol. Column generation techniques for pickup and delivery problems. Ph.D. thesis, Eindhoven University of Technology, Netherlands, 1994.

[49] R.S. Solanki and F. Southworth. An execution planning algorithm for military airlift. *Interfaces*, 21:121–131, 1991.

[50] M.M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35:254–265, 1987.

[51] M.M. Solomon, A. Chalifour, J. Desrosiers, and J. Boisvert. An application of vehicle routing methodology to large-scale larvicide control programs. *Interfaces*, 22:88–99, 1992.

[52] M.M. Solomon and J. Desrosiers. Time window constrained routing and scheduling problems. *Transportation Science*, 22:1–13, 1988.

[53] D.M. Stein. Scheduling dial-a-ride transportation systems. *Transportation Science*, 12:232–249, 1978.

[54] P.M. Thompson and H.N. Psaraftis. Cyclic transfer algorithms for multi-vehicle routing and scheduling problems. *Operations Research*, 41:935–946, 1993.

[55] P. Toth and D. Vigo. Fast local search algorithms for the handicapped persons transportation problem. In I.H. Osman and J.P. Kelly, editors, *Metaheuristics: Theory and Applications*, Kluwer, Boston, MA, 1996.

[56] P. Toth and D. Vigo. Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, 31:60–71, 1997.

[57] 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*, 27:298–311, 1993.

[58] H. Wilson and N. Colvin. Computer control of the Rochester dial-a-ride system. Technical Report R-77-31, Department of Civil Engineering, MIT, Cambridge, MA, 1977.

[59] H. Wilson, J. Sussman, H. Wang, and B. Higonnet. Scheduling algorithms for dial-a-ride systems. Technical Report USL TR-70-13, Urban Systems Laboratory, MIT, Cambridge, MA, 1971.

[60] H. Wilson and H. Weissberg. Advanced dial-a-ride algorithms research project: Final report. Technical Report R76-20, Department of Civil Engineering, MIT, Cambridge, MA, 1976.