Chapter 3

VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Brian Kallehauge
Jesper Larsen
Oli B.G. Madsen
Marius M. Solomon

Abstract In this chapter we discuss the Vehicle Routing Problem with Time Windows in terms of its mathematical modeling, its structure and decomposition alternatives. We then present the master problem and the subproblem for the column generation approach, respectively. Next, we illustrate a branch-and-bound framework and address acceleration strategies used to increase the efficiency of branch-and-price methods. Then, we describe generalizations of the problem and report computational results for the classic Solomon test sets. Finally, we present our conclusions and discuss some open problems.

1. Introduction

The vehicle routing problem (VRP) involves finding a set of routes, starting and ending at a depot, that together cover a set of customers. Each customer has a given demand, and no vehicle can service more customers than its capacity permits. The objective is to minimize the total distance traveled or the number of vehicles used, or a combination of these. In this chapter, we consider the vehicle routing problem with time windows (VRPTW), which is a generalization of the VRP where the service at any customer starts within a given time interval, called a time window. Time windows are called soft when they can be considered non-biding for a penalty cost. They are hard when they cannot be violated, i.e., if a vehicle arrives too early at a customer, it must wait until the time window opens; and it is not allowed to arrive late. This is the case we consider here.

The remarkable advances in information technology have enabled companies to focus on efficiency and timeliness throughout the supply chain. In turn, the VRPTW has increasingly become an invaluable tool in modeling a variety of aspects of supply chain design and operation. Important VRPTW applications include deliveries to supermarkets, bank and postal deliveries, industrial refuse collection, school bus routing, security patrol service, and urban newspaper distribution. Its increased practical visibility has evolved in parallel with the development of broader and deeper research directed at its solution. Significant progress has been made in both the design of heuristics and the development of optimal approaches.

In this chapter we will concentrate on exact methods for the VRPTW based on column generation. These date back to Desrochers, Desrosiers and Solomon (1992) who used column generation in a Dantzig-Wolfe decomposition framework and Halse (1992) who implemented a decomposition based on variable splitting (also known as Lagrangean decomposition). Later, Kohl and Madsen (1997) developed an algorithm exploiting Lagrangean relaxation. Then, Kohl, Desrosiers, Madsen, Solomon and Soumis (1999); Larsen (1999); Cook and Rich (1999) extended the previous approaches by developing Dantzig-Wolfe based decomposition algorithms involving cutting planes and/or parallel platforms. Kallehauge (2000) suggested a hybrid algorithm based on a combination of Lagrangean relaxation and Dantzig-Wolfe decomposition. Recently, Chabrier (2005); Chabrier, Danna and Le Pape (2002); Feillet, Dejax, Gendreau and Gueguen (2004); Irnich and Villeneuve (2005); Rousseau, Gendreau and Pesant (2004) have proposed algorithms based on enhanced subproblem methodology. Advancements in master problem approaches have been made by Danna and Le Pape (2005); Larsen (2004).

This chapter has the following organization. In Section 2 we describe the mathematical model of the VRPTW and in Section 3 we discuss the structure of the problem and decomposition alternatives. Next, Sections 4 and 5 present the master problem and the subproblem for the column generation approach, respectively. Section 6 illustrates the branch-and-bound framework, while Section 7 addresses acceleration strategies used to increase the efficiency of branch-and-price methods. Then, we describe generalizations of the VRPTW in Section 8 and report computational results for the classic Solomon test sets in Section 9. Finally we present our conclusions and discuss some open problems in 10.

2. The model

The VRPTW is defined by a fleet of vehicles, $\mathcal{V}$, a set of customers, $\mathcal{C}$, and a directed graph $\mathcal{G}$. Typically the fleet is considered to be homogeneous, that is, all vehicles are identical. The graph consists of $|\mathcal{C}| + 2$ vertices, where the customers are denoted $1, 2, \ldots, n$ and the depot is represented by the vertices 0 ("the starting depot") and $n + 1$ ("the returning depot"). The set of all vertices, that is, $0, 1, \ldots, n+1$ is denoted $\mathcal{N}$. The set of arcs, $\mathcal{A}$, represents direct connections between the depot and the customers and among the customers. There are no arcs ending at vertex 0 or originating from vertex $n + 1$. With each arc (i, j), where $i \neq j$, we associate a *cost* c_{ij} and a *time* t_{ij}, which may include service time at customer i.

Each vehicle has a capacity q and each customer i a demand d_i. Each customer i has a *time window* $[a_i, b_i]$ and a vehicle must arrive at the customer before b_i. If it arrives before the time window opens, it has to wait until a_i to service the customer. The time windows for both depots are assumed to be identical to $[a_0, b_0]$ which represents the *scheduling horizon*. The vehicles may not leave the depot before a_0 and must return at the latest at time b_{n+1}.

It is assumed that q, a_i, b_i, d_i, c_{ij} are non-negative integers and t_{ij} are positive integers. Note that this assumption is necessary to develop an algorithm for the shortest path with resource constraints used in the column generation approach presented later. Furthermore it is assumed that the triangle inequality is satisfied for both c_{ij} and t_{ij}.

The model contains two sets of decision variables x and s. For each arc (i, j), where $i \neq j$, $i \neq n + 1$, $j \neq 0$, and each vehicle k we define x_{ijk} as

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ drives directly from vertex } i \text{ to vertex } j, \\ 0, & \text{otherwise.} \end{cases}$$

The decision variable s_{ik} is defined for each vertex i and each vehicle k and denotes the time vehicle k starts to service customer i. In case vehicle k does not service customer i, s_{ik} has no meaning and consequently it's value is considered irrelevant. We assume $a_0 = 0$ and therefore $s_{0k} = 0$, for all k.

The goal is to design a set of routes that minimizes total cost, such that

- each customer is serviced exactly once,
- every route originates at vertex 0 and ends at vertex $n + 1$, and

- the time windows of the customers and capacity constraints of the vehicles are observed.

This informal VRPTW description can be stated mathematically as a multicommodity network flow problem with time windows and capacity constraints:

$$\min \sum_{k\in\mathcal{V}} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} c_{ij} x_{ijk} \quad s.t., \tag{3.1}$$

$$\sum_{k\in\mathcal{V}} \sum_{j\in\mathcal{N}} x_{ijk} = 1 \quad \forall i \in \mathcal{C}, \tag{3.2}$$

$$\sum_{i\in\mathcal{C}} d_i \sum_{j\in\mathcal{N}} x_{ijk} \leq q \quad \forall k \in \mathcal{V}, \tag{3.3}$$

$$\sum_{j\in\mathcal{N}} x_{0jk} = 1 \quad \forall k \in \mathcal{V}, \tag{3.4}$$

$$\sum_{i\in\mathcal{N}} x_{ihk} - \sum_{j\in\mathcal{N}} x_{hjk} = 0 \quad \forall h \in \mathcal{C},\ \forall k \in \mathcal{V}, \tag{3.5}$$

$$\sum_{i\in\mathcal{N}} x_{i,n+1,k} = 1 \quad \forall k \in \mathcal{V}, \tag{3.6}$$

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0 \quad \forall i,j \in \mathcal{N},\ \forall k \in \mathcal{V}, \tag{3.7}$$

$$a_i \leq s_{ik} \leq b_i \quad \forall i \in \mathcal{N},\ \forall k \in \mathcal{V}, \tag{3.8}$$

$$x_{ijk} \in \{0,1\} \quad \forall i,\ j \in \mathcal{N},\ \forall k \in \mathcal{V}. \tag{3.9}$$

The objective function (3.1) minimizes the total travel cost. The constraints (3.2) ensure that each customer is visited exactly once, and (3.3) state that a vehicle can only be loaded up to it's capacity. Next, equations (3.4), (3.5) and (3.6) indicate that each vehicle must leave the depot 0; after a vehicle arrives at a customer it has to leave for another destination; and finally, all vehicles must arrive at the depot $n+1$. The inequalities (3.7) establish the relationship between the vehicle departure time from a customer and its immediate successor. Finally constraints (3.8) affirm that the time windows are observed, and (3.9) are the integrality constraints. Note that an unused vehicle is modeled by driving the "empty" route $(0, n+1)$.

The model can also incorporate a constraint giving an upper bound on the number of vehicles, as is the case in Desrosiers, Dumas, Solomon and Soumis (1995):

$$\sum_{k\in\mathcal{V}} \sum_{j\in\mathcal{N}} x_{0jk} \leq |V| \quad \forall k \in \mathcal{V},\ \forall j \in \mathcal{N} \tag{3.10}$$

Note also that the nonlinear restrictions (3.7) can be linearized as:

$$s_{ik} + t_{ij} - M_{ij}(1 - x_{ijk}) \leq s_{jk} \quad \forall i,\ j \in \mathcal{N},\ \forall k \in \mathcal{V}. \tag{3.11}$$

The large constants M_{ij} can be decreased to $\max\{b_i + t_{ij} - a_j\}$, $(i, j) \in A$.

For each vehicle, the service start variables impose a unique route direction thereby eliminating any subtours. Hence, the classical VRP subtour elimination constraints become redundant. Finally, the objective function (3.1) has been universally used when solving the VRPTW to optimality. In the research on heuristics it has been common to minimize the number of vehicles which may lead to additional travel cost.

The VRPTW is a generalization of both the traveling salesman problem (TSP) and the VRP. When the time constraints (3.7) and (3.8)) are not binding the problem relaxes to a VRP. This can be modeled by setting $a_i = 0$ and $b_i = M$, where M is a large scalar, for all customers i. If only one vehicle is available the problem becomes a TSP. If several vehicles are available and the cost structure is: $c_{0j} = 1$, $j \in \mathcal{C}$ and $c_{ij} = 0$, otherwise, we obtain the bin-packing problem. Since trips between customers are "free", the order in which these are visited becomes unimportant and the objective turns to "squeezing" as much demand as possible into as few vehicles (bins) as possible. In case the capacity constraints (3.2) are not binding the problem becomes a m-TSPTW, or, if only one vehicle is available, a TSPTW.

3. Structure and decomposition

A closer look at the above model reveals that only the assignment constraints (3.2) are coupling the vehicles while the remaining constraints are dealing with each vehicle separately. This strongly suggests the use of Lagrangean relaxation (LR) or decomposition, for example Dantzig-Wolfe (DWD), to break up the overall problem into a subproblem for each vehicle and a master problem. To date, the most successful decomposition approaches for the VRPTW cast the subproblem as a constrained shortest path structure. The master problem is an integer program whose solution cannot be obtained directly, so its LP relaxation is solved. The column generation process alternates between solving this linear master problem and the subproblem. The former finds new multipliers to send to the latter which uses this information to find new columns to send back. A lower bound on the optimal integer solution of the VRPTW model is obtained at the end of this back and forth process. This is then used within a branch-and-bound framework to obtain the optimal VRPTW solution. If the vehicles are identical, as we have assumed here, all subproblems will be equivalent and therefore it is necessary to only solve one. The master problem and the subproblem will be discussed in more detail in Sections 4 and 5, respectively. The complete column generation process is described in Section 1, while the subproblem forms the subject of Section 2.

In addition, other LRs are possible but not promissing. One may consider relaxing the time and capacity constraints (3.3), (3.7) and (3.8). This yields a linear network flow problem which possesses the integrality property. The corresponding bound can be calculated very fast, but is not likely to be very strong unless capacity is not binding and time windows are very narrow (see Desrosiers, Dumas, Solomon and Soumis, 1995). Relaxing only the capacity or time window constraints also does not seem sensible since the relaxed problem is not generally easier to solve than the original.

Desrochers, Desrosiers and Solomon (1992) were the first to apply DWD with a free number of vehicles. The assignment constraints were considered the coupling constraints, while the subproblem was a shortest path problem with resource constraints. Relaxing the same constraint set and applying LR was first proposed by Kohl and Madsen (1997). Desrosiers, Sauvé and Soumis (1988) have used a similar relaxation to calculate a lower bound for the minimum fleet size for the m-TSPTW.

Jörnsten, Madsen and Sørensen (1986) suggested solving the VRPTW by variable splitting (later called Lagrangean decomposition, or LD). In follow-up work, Halse (1992) described three different variable splitting methods where $\sum_j x_{ijk}$ was replaced by y_{ik} in constraint set (3.2) and possibly (3.3). In turn, the constraint $y_{ik} = \sum_j x_{ijk}$ was introduced and Lagrangeanly relaxed. The problem decomposes into two problems, one in the x- and s-variables and the other in the y-variables. The former problem is further decomposed by vehicle and it is a shortest path problem with resource constraints. The latter is an assignment-type problem. Specifically, the approaches are:

- VS1: Keep constraints (3.2) and (3.3) in the y-problem. This represents a generalized assignment problem (GAP) and the x/s-problem becomes a shortest path problem with time windows (SPPTW). The GAP has the special structure where all right hand sides in (3.3) are identical and d_i does not depend on k.

- VS2: Keep constraints (3.2) in the y-problem. The y-problem becomes a "Semi assignment" problem (SAP) consisting of constraints (3.2) only. The x/s-problem is equivalent to a shortest path problem with time windows and capacity constraints (SPPTWCC). The SAP is easily solvable and possesses the integrality property.

- VS3: Keep constraints (3.2) in the y-problem and constraints (3.3) in both the y- and the x/s-problem. The y-problem is a GAP and the x/s-problem constitutes a SPPTWCC.

In the LD master problem, whose role is to find multipliers to the relaxed equation relating x and y, the number of multipliers is larger than in the LR considered above. This clearly makes the master problem more difficult. Also the subproblems are no longer identical since the LD multipliers depend on both customer and vehicle. Note that only VS1 and VS2 have been implemented.

We now define LB(VS1), LB(VS2) and LB(VS3) as the best lower bounds obtainable from the three variable splitting approaches, respectively. It can be shown that the previous LR and the DWD yield the same lower bound LB(LR/DWD). Provided that the vehicles are identical, Kohl (1995) has derived the following results:

$$\begin{aligned}
\text{LB(VS3)} &\geq \text{LB(VS1)}, \\
\text{LB(VS3)} &\geq \text{LB(VS2)}, \\
\text{LB(LR/DWD)} &= \text{LB(VS2)}.
\end{aligned}$$

There exist instances for which LB(VS3) $>$ LB(VS1). He further showed that LB(VS2) $=$ LB(VS3) under some weak supplementary conditions. This is surprising because it implies there is no additional gain to be derived from solving two hard integer problems (the SPPTWCC and GAP) instead of just one (the SPPTWCC). However, in the more general case where vehicles have different capacities it might be possible that the VS3 model yields a better bound than VS2.

To conclude, in VRPTW case, the variable splitting methods mentioned above generally provide similar lower bounds to those obtained from the ordinary LR or DWD.

4. The master problem

The column generation methodology has been successfully applied to the VRPTW by numerous researchers. It represents a generalization of the linear DWD since the master problem and the subproblem are integer and mixed-integer programs, respectively. Often the master problem is simply stated as a set partitioning problem on which column generation is applied, thereby avoiding the description of the DWD on which it is based. To gain an appreciation for different cutting and branching opportunities compatible with column generation, here we present the master problem by going through the steps of the DWD based on the multicommodity network flow formulation (3.1)–(3.9).

The column generation approach exploits the fact that only constraint set (3.2) links the vehicles together. Hence, the integer master problem is defined through (3.1)–(3.2) and (3.9), that is, it contains the objective function, the assignment of customers to exactly one vehicle and the binary requirement on the flow variables. The rest of the constraints and (3.9) are part of the subproblem which has a modified objective function that decomposes into $|V|$ independent subproblems, one for each vehicle. In the rest of this section we will focus on the linear master problem (3.1)–(3.2). Branching, necessary to solve the integer master problem, will be discussed in Section 6.

Let $\mathcal{P}^k$ be the set of feasible paths for vehicle k, $k \in \mathcal{V}$. Hence, $p \in \mathcal{P}^k$ corresponds to an elementary path which can also be described by using the binary values x^k_{ijp}, where $x^k_{ijp} = 1$, if vehicle k goes directly from vertex i to vertex j on path p, and $x^k_{ijp} = 0$, otherwise. Any solution x^k_{ij} to the master problem (3.1)–(3.2) can be written as a non-negative convex combination of a finite number of elementary paths, i.e.,

$$x^k_{ij} = \sum_{p \in \mathcal{P}^k} x^k_{ijp} y^k_p \quad \forall k \in \mathcal{V},\ \forall (i,j) \in \mathcal{A}, \tag{3.12}$$

$$\sum_{p \in \mathcal{P}^k} y^k_p = 1 \quad \forall k \in \mathcal{V}, \tag{3.13}$$

$$y^k_p \geq 0 \quad \forall k \in \mathcal{V},\ \forall p \in \mathcal{P}^k. \tag{3.14}$$

Using x^k_{ijp} we can define the cost of a path, c^k_p, and the number of times a customer i is visited by vehicle k, a^k_i, as:

$$c^k_p = \sum_{(i,j) \in \mathcal{A}} c^k_{ij} x^k_{ijp} \quad \forall k \in \mathcal{V},\ \forall p \in \mathcal{P}^k,$$

$$a^k_{ip} = \sum_{j \in \mathcal{N} \cup \{n+1\}} x^k_{ijp} \quad \forall k \in \mathcal{V},\ \forall i \in \mathcal{N},\ \forall p \in \mathcal{P}^k.$$

Now we can substitute these values into (3.1)–(3.2) and arrive at the revised formulation of the master problem:

$$\min \sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}^k} c^k_p y^k_p \ \ s.t., \tag{3.15}$$

$$\sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}^k} a^k_{ip} y^k_p = 1 \quad \forall i \in \mathcal{C}, \tag{3.16}$$

$$\sum_{p \in \mathcal{P}^k} y^k_p = \quad 1 \quad \forall k \in \mathcal{V}, \tag{3.17}$$

$$y_p^k \geq 0 \quad \forall k \in \mathcal{V},\ \forall p \in \mathcal{P}^k. \tag{3.18}$$

The mathematical formulation (3.15)–(3.18) is then the linear relaxation of a set partitioning type problem with an additional constraint on the total number of vehicles and a set of convex combination constraints.

In the usual case of a single depot and a homogeneous fleet of vehicles with the same initial conditions for all vehicles, all $\mathcal{P}^k$ are identical, that is, $\mathcal{P}^k = \mathcal{P}$, $k \in \mathcal{V}$. Furthermore, the networks for the subproblems are also identical. Therefore constraints (3.17) can be aggregated. By letting $y_p = \sum_{k\in\mathcal{V}} y_p^k$, the index k can be eliminated from the formulation (3.15)–(3.18). The resulting model given below is the classical linear relaxation of the set partitioning formulation:

$$\min \sum_{p\in\mathcal{P}} c_p y_p \ \ s.t., \tag{3.19}$$

$$\sum_{p\in\mathcal{P}} a_{ip} y_p = 1 \quad \forall i \in \mathcal{C}, \tag{3.20}$$

$$y_p \geq 0 \quad \forall p \in \mathcal{P}. \tag{3.21}$$

In the column generation methodology, the set of columns in the linear master problem is limited to only those that have already been generated, hence the term *restricted* master problem. It consists of finding a set of minimum cost paths among all paths presently in the master problem. The restricted master problem can mathematically be stated as:

$$\min \sum_{p\in\mathcal{P}'} c_p y_p \ \ s.t., \tag{3.22}$$

$$\sum_{p\in\mathcal{P}'} a_{ip} y_p = 1 \quad \forall i \in \mathcal{C}, \tag{3.23}$$

$$y_p \geq 0 \quad \forall p \in \mathcal{P}'. \tag{3.24}$$

Each decision variable y_p counts the number of times path p is used. This is not necessarily integer, but can be any real number in the interval $[0; 1]$. The set $\mathcal{P}'$ contains all the paths generated, a_{ip} denotes the number of times customer i is serviced on path p, and, c_p is the cost of the path. The parameter a_{ip} should in principle be either 0 or 1, but since the subproblem is relaxed (see Section 5) it can take larger integer values.

Solving the restricted master yields a solution $y = (y_1, y_2, \ldots, y_{|\mathcal{P}'|})$ which might be integer but this is not guaranteed. If it is integer, a feasible but not necessarily optimal solution to the VRPTW has been found. In addition to the primal solution, a dual solution $\phi = (\phi_1, \phi_2, \ldots, \phi_{|C|})$ is also obtained.

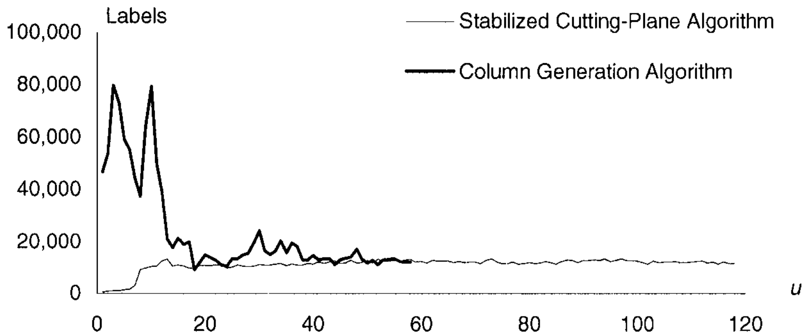
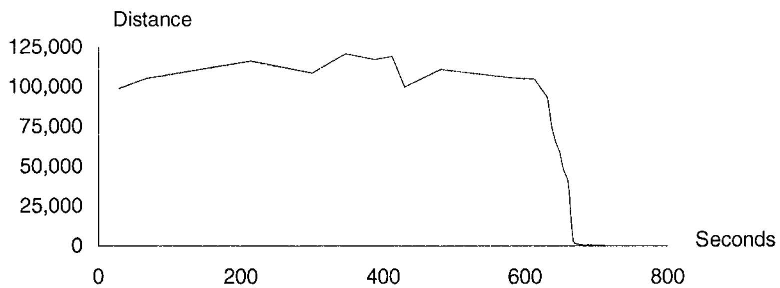
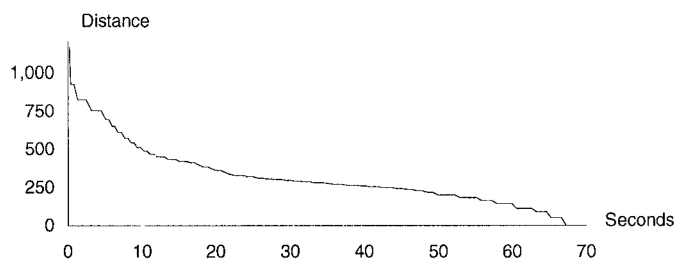


Figure 3.1. Number of labels generated in the subproblem wrt. the iteration number for the Dantzig-Wolfe method and the bundle method on the Solomon instance R104 with 100 customers (from Kallehauge, 2000).



(a) Column generation algorithm.



(b) Stabilized cutting-plane algorithm.

Figure 3.2. The Euclidian distance between the current dual variables and the optimum dual variables. Observe the different scales.

An initial start for the restricted master problem is often the set of routes visiting a single customer, that is, routes of the type depot-i-depot (cf. Section 8). When the optimal solution to the restricted master problem is found, the simplex algorithm asks for a new variable (i.e. a column/path $p \in \mathcal{P} \setminus \mathcal{P}'$) with negative reduced cost. Such a column is found by solving a subproblem, sometimes called the pricing problem. For the VRPTW, the subproblem should solve the problem "Find the path with minimal reduced cost." Solving the subproblem is in fact an implicit enumeration of all feasible paths, and the process terminates when the optimal objective of the subproblem is non-negative (it will actually be 0).

It is not surprising that the behavior of the dual variables plays a pivotal role in the overall performance of the column generation principle for the VRPTW. It has been observed by Kallehauge (2000) that dual variables do not converge smoothly to their respective optima. Assume that the paths $(0, i, n+1)$ are used to initialize the algorithm. Figure 3.1 illustrates the instability of the column generation algorithm compared to the stabilized cutting-plane algorithm presented in the above paper. Furthermore, Figure 3.2 illustrates the effect of the size of the multipliers on the computational difficulty of the SPPTWCC subproblems. Whereas the multipliers are large in the Dantzig-Wolfe process, they are small in the cutting-plane approach. This problem originates in the coordination between the master problem and the subproblem.

Finally, in many routing problems the optimal solution remains unchanged even if overcovering rather than exact covering of customers is allowed. Due to the triangle inequality in the VRPTW, overcovering will always be more expensive than just covering and therefore an optimal solution will always be one where each customer is visited exactly once. The advantage of allowing overcovering is that the linear relaxation of the Set Covering Problem is easier to solve than that of the Set Partitioning Problem, and this will in turn lead to the computation of good estimates of the dual variables.

5. The subproblem

In the column generation approach for the VRPTW, the subproblem decomposes into $|\mathcal{V}|$ identical problems, each one being a shortest path problem with resource constraints (time windows and vehicle capacity). More specifically, the subproblem is an Elementary Shortest Path Problem with Time Windows and Capacity Constraints (ESPPTWCC), where elementary means that each customer can appear at most once in the shortest path. It can be formulated as:

$$\min \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \hat{c}_{ij} x_{ij}, \ \ s.t., \tag{3.25}$$

$$\sum_{i \in \mathcal{C}} d_i \sum_{j \in \mathcal{N}} x_{ij} \leq q, \tag{3.26}$$

$$\sum_{j \in \mathcal{N}} x_{0j} = 1, \tag{3.27}$$

$$\sum_{i \in \mathcal{N}} x_{ih} - \sum_{j \in \mathcal{N}} x_{hj} = 0 \quad \forall h \in \mathcal{C}, \tag{3.28}$$

$$\sum_{i \in \mathcal{N}} x_{i,n+1} = 1, \tag{3.29}$$

$$s_i + t_{ij} - M_{ij}(1 - x_{ij}) \leq s_j \quad \forall i, \ j \in \mathcal{N}, \tag{3.30}$$

$$a_i \leq s_i \leq b_i \quad \forall i \in \mathcal{N}, \tag{3.31}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, \ j \in \mathcal{N} \tag{3.32}$$

Constraint (3.26) is the capacity constraint, constraints (3.30) and (3.31) are time constraints, while constraint (3.32) ensures integrality. The constraints (3.27), (3.28) and (3.29) are flow constraints resulting in a path from the depot 0 to the depot $n + 1$. When solving the ESPPTWCC as the subproblem in the VRPTW, $\hat{c}_{ij}$ is the *modified cost* of using arc (i, j), where $\hat{c}_{ij} = c_{ij} - \pi_i$. Note that while c_{ij} is a non-negative integer, $\hat{c}_{ij}$ can be any real number.

This subproblem does not posses the integrality property, and therefore solving it as a linear mixed-integer programming problem will potentially result in a reduction of the integrality gap between the optimal solution of the LP-relaxed version of the VRPTW and the optimal integer solution to the problem.

Since the ESPPTWCC is NP-hard in the strong sense (see Dror, 1994; Kohl, 1995), the usual approach has been to slightly alter the problem by relaxing some of the constraints. In particular, allowing cycles changes the problem to the Shortest Path Problem with Time Windows and Capacity Constraints (SPPTWCC). Since arcs can now be used more than once (and customers may therefore be visited more than once), the decision variables x_{ij} and s_i are replaced by x^l_{ij} and s^l_i. The variable x^l_{ij} is set to 1 if the arc (i, j) is used as the l'th arc on the shortest path, and 0 otherwise, and the variable s^l_i is set to the start of service at customer i as customer number l, where $l \in \mathcal{L} = \{1, 2, \ldots, |\mathcal{L}|\}$, $|\mathcal{L}| = \lfloor b_{n+1} / \min t_{ij} \rfloor$. The SPPTWCC can now be described by the following mathematical model:

$$\min \sum_{l\in\mathcal{L}} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \hat{c}_{ij} x_{ij}^l, \quad s.t. \tag{3.33}$$

$$\sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} x_{ij}^1 = 1, \tag{3.34}$$

$$\sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} x_{ij}^l - \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} x_{ij}^{l-1} \leq 0 \quad \forall l \in \mathcal{L} - \{1\}, \tag{3.35}$$

$$\sum_{i\in\mathcal{C}} d_i \sum_{l\in\mathcal{L}} \sum_{j\in\mathcal{N}} x_{ij}^l \leq q, \tag{3.36}$$

$$\sum_{j\in\mathcal{N}} x_{0j}^1 = 1, \tag{3.37}$$

$$\sum_{i\in\mathcal{N}} x_{ih}^{l-1} - \sum_{j\in\mathcal{N}} x_{hj}^l = 0 \quad \forall h \in \mathcal{C}\ \forall l \in \mathcal{L} - \{1\}, \tag{3.38}$$

$$\sum_{l\in\mathcal{L}} \sum_{i\in\mathcal{N}} x_{i,n+1}^l = 1, \tag{3.39}$$

$$s_i^l + t_{ij} - K(1 - x_{ij}^l) \leq s_j^l \quad \forall i,\ j \in \mathcal{N}\ \forall l \in \mathcal{L} - \{1\}, \tag{3.40}$$

$$a_i \leq s_i^l \leq b_i \quad \forall i \in \mathcal{N}, \tag{3.41}$$

$$x_{ij}^l \in \{0,1\} \quad \forall i,j \in \mathcal{N}. \tag{3.42}$$

In this formulation, (3.34) forces the first arc to be used only once, while (3.35) states that arc l can only be used provided that arc $l-1$ is used. The remaining constraints are the original constraints (3.3) to (3.9) extended to include the additional superscript l and the changes related to its inclusion. Note that (3.34) is redundant as it is covered by (3.37), but it has been kept in the model as to indicate the origin node.

This problem can be solved by a pseudo-polynomial algorithm described in Desrochers, Desrosiers and Solomon (1992). This and all other current approaches are based on dynamic programming. Even though negative cycles are possible, the time windows and the capacity constraints prohibits infinite cycling. Note that capacity is accumulated every time a customer is serviced in a cycle. If the distance used to compute the cost of routes satisfies the triangle inequality, the optimal solution contains only elementary routes. Solving the SPPTWCC instead of the ESPPTWCC augments the size of the set of admissible columns generated for the master problem. Consequently the lower bound on the master problem may decrease. A slight improvement can be obtained by implementing 2-cycle elimination in the solution process which dates back to Kolen, Rinnooy Kan and Trienekens (1987).

While the SPPTWCC relaxation was at the time a computational necessity, the ESPPTWCC has recently been tackled directly. Work on this problem and k-cycle elimination, where $k \geq 3$, proved very successful in expanding the scope of the VRPTW problems solved. Even though the ESPPTWCC continues to be regarded as difficult to solve when time windows are wide, two research groups have recently used it directly in VRPTW optimal algorithms. Chabrier (2005); Chabrier, Danna and Le Pape (2002), and independently Feillet, Dejax, Gendreau and Gueguen (2004) have extended the dynamic programming approach of Desrochers, Desrosiers and Solomon (1992) to the ESPPTWCC by adapting the path dominance rule. They then incorporated several heuristic modifications to make the algorithm much faster. Chabrier (2005) and Chabrier, Danna and Le Pape (2002) obtained lower bounds superior to those based on the SPPTWCC resulting in excellent computational results to be described in Section 9. A different approach that has not yet been tried on the VRPTW is presented in Dumitrescu and Boland (2003). The authors compare three scaling techniques and a standard label-setting method. They show that integrating preprocessing information within the label-setting method can be very beneficial in terms of both memory and run time. Further improvements of the label-setting method can be obtained by using Lagrangean relaxation.

Instead of dealing with the computational burden of the ESPPTWCC or the weakened lower bound provided by the SPPTWCC, one could consider a middle of the road approach. That is, disallow cycles of small length. As discussed above, cycle elimination corresponding to $k = 2$ has been a common technique. In the SPPTWCC-k-cyc, paths with cycles of length of at most k are eliminated. The case $k \geq 3$ has been considered by Irnich and Villeneuve (2005) with encouraging results presented in Section 9. Recently Rousseau, Gendreau and Pesant (2004) have presented results where Constraint Programming is used to solve the subproblem. Taking into account the difference in computer power, the authors conclude that their approach is not any faster than that of Desrochers, Desrosiers and Solomon (1992).

6. Branch-and-bound

The column generation approach does not automatically guarantee integer solutions and often the solutions obtained will indeed be fractional. Therefore a branch-and-bound framework has to be established. The calculations are organised in a branching tree. For the VRPTW only binary strategies have been proposed in the literature although it should be noted that it is generally not difficult to come up with non-binary branching trees for the problem. The branching decisions are generally based on considerations related to the original 3-index flow formulation (3.2)–(3.9). The column generation process is then repeated at each node in the branch-and-bound tree.

6.1 Branching on the number of vehicles

Branching on the number of vehicles was originally proposed by Desrochers, Desrosiers and Solomon (1992). If the number of vehicles is fractional we introduce a bound on the number of vehicles. Note that this branching strategy does not require that the flow and time variables of the original model be computed.

This branching rule can be implemented fairly easily and only concerns the master problem. We denote the flow over an arc by f_{ij} and this is the sum of all flows over that arc, that is $f_{ij} = \sum_{k \in \mathcal{V}} x_{ijk}$. The f_{ij} values can easily be derived from the solution of the master problem. When we branch on the number of vehicles, two child nodes are created, one imposing on the master problem parent node the additional constraint $\sum_{j \in C} f_{0j} \geq \lceil l \rceil$ while the other forcing $\sum_{j \in C} f_{0j} \leq \lfloor l \rfloor$, where l is the fractional sum of all variables in the master problem.

Note that branching on the number of vehicles is not necessarily enough to obtain an integer solution as it is possible to derive solutions where the sum of the vehicles is integer, but yet there are fractional vehicles driving around the network.

6.2 Branching on flow variables

Branching on a single variable x_{ijk} is possible only if each vehicle can be distinguished. In column generation this can be achieved by solving the subproblem for *each* vehicle individually and by introducing an additional constraint in the master problem

$$\sum_{p \in P_k} y_p = 1 \quad \forall k \in \mathcal{V}$$

where P_k is the set of routes generated for each vehicle k and y_p is the binary variable indicating whether route p is used.

Since most cases described in the literature assume a homogeneous fleet, it doesn't make sense to branch on individual vehicles. Instead, branching can be done on sums of flows, that is either on $\sum_j x_{ijk}$ or on $\sum_k x_{ijk}$ (equivalent to f_{ij}). Branching on $\sum_j x_{ijk}$ results in a different subproblem for each vehicle, even though the vehicles are identical. That is because imposing $\sum_j x_{ijk} = 1$ forces customer i to be visited by vehicle

k, while $\sum_j x_{ijk} = 0$ implies that customer i is assigned to any vehicle but k.

The standard practice has been to branch on $\sum_k x_{ijk}$ since the branching decision can easily be transfered to the master problem and subproblem. This was proposed independently by Halse (1992); Desrochers, Desrosiers and Solomon (1992). When $\sum_k x_{ijk} = 1$, customer j succeeds customer i on the same route, while if $\sum_k x_{ijk} = 0$, customer i does not immediately precede j. If there is more than one candidate for branching, that is, there are several fractional variables, we would generally like to choose a candidate that is not close to either 0 or 1 in order to make an impact. When selecting among the nodes to branch on, a common heuristic is to branch on the variable maximizing $c_{ij}(\min\{x_{ijk}, 1 - x_{ijk}\})$ using a best-first strategy In order to create more complex strategies the branching schemes can be applied hierarchically, such as first branching on the number of vehicles and then on $\sum_k x_{ijk}$, or mixed.

6.3 Branching on resource windows

Branching on resource windows was first proposed by Gélinas, Desrochers, Desrosiers and Solomon (1995) and is presently the only alternative to branching on flow variables. In the VRPTW model resource windows can be interpreted as either the time windows or the capacity constraints. We will only discuss branching on time windows, as capacity is significantly less constraining in many cases. In Gélinas, Desrochers, Desrosiers and Solomon (1995) only branching on time windows was used.

Branching on time windows results in splitting a time window into two smaller ones. Branching has to be done in such a way that at least one route is infeasible in each of the two sub-windows.

In order to branch on time windows three decisions have to be taken:

1) How should the node for branching be chosen?

2) Which time window should be divided?

3) Where should the partition point be?

In order to decide on the above issues, we define *feasibility intervals* $[l_i^r, u_i^r]$ for all vertices $i \in \mathcal{N}$ and all routes r with fractional flow. l_i^r is the earliest time that service can start at vertex i on route r, and u_i^r is the latest time that service can start, that is, $[l_i^r, u_i^r]$ is the time interval during which route r must visit vertex i to remain feasible.

The intervals can easily be computed by a recursive formula. Additionally we define

$$L_i = \max_{\text{fractional routes } r} \{l_i^r\}, \quad i \in \mathcal{N}, \tag{3.43}$$

$$U_i = \min_{\text{fractional routes } r} \{u_i^r\}, \quad i \in \mathcal{N}. \tag{3.44}$$

If $L_i > U_i$ at least two routes (or two visits by the same route) have disjoint feasibility intervals, i.e., the vertex is a candidate for branching on time windows. We can branch on a candidate vertex i by dividing the time windows $[a_i, b_i]$ at any integer value in the open interval $[U_i, L_i[$. It should be noted that situations can arise where there are no candidates for branching on time windows, but the solution is not feasible.

Three different strategies were proposed by Gélinas, Desrochers, Desrosiers and Solomon (1995) aiming at the elimination of cycles, the minimization of the number of visits to a customer i and the balancing of flow in the two branch-and-bound nodes.

After having chosen the candidate vertex i for branching, an integer $t \in [U_i, L_i[$ has to be selected in order to determine the division. Here t is chosen in order to divide the time window of the customer such that 1) the flow is balanced and 2) the time window is divided as evenly as possible.

7. Acceleration strategies

7.1 Preprocessing

The aim of preprocessing is to narrow the solution space by tightening the formulation before the actual optimization is started. This can be done by fixing some variables, reducing the interval of values a variable can take and so on. In the VRPTW, the time windows can be narrowed if the triangle inequality holds. Accordingly, Kontoravdis and Bard (1995) propose the following scheme. The earliest time a vehicle can arrive at a customer is by arriving straight from the depot and the latest time it can leave is by going directly back to the depot. Hence, for each customer i, its time window can be strengthened from $[a_i, b_i]$ to $[\max\{a_0 + t_{0i}, a_i\}, \min\{b_{n+1} - t_{i,n+1}, b_i\}]$.

A further reduction of the time windows can be achieved by the method developed by Desrochers, Desrosiers and Solomon (1992). The time windows are reduced by applying the following four rules in a cyclic manner. The process is stopped when one whole cycle is performed without changing any of the time windows. The four rules are:

1) Minimal arrival time from predecessors:

$$a_l = \max\left\{a_l, \min\left\{b_l, \min_{(i,l)}\{a_i + t_{il}\}\right\}\right\}.$$

2) Minimal arrival time to successors:

$$a_l = \max\left\{a_l, \min\left\{b_l, \min_{(l,j)}\{a_j - t_{lj}\}\right\}\right\}.$$

3) Maximal departure time from predecessors:

$$b_l = \min\left\{b_l, \max\left\{a_l, \max_{(i,l)}\{b_i + t_{il}\}\right\}\right\}.$$

4) Maximal departure time to successors:

$$b_l = \min\left\{b_l, \max\left\{a_l, \max_{(l,j)}\{b_j - t_{lj}\}\right\}\right\}.$$

The first rule adjusts the start of the time window to the earliest time a vehicle can arrive coming straight from any possible predecessor. In a similar fashion, the second rule modifies the start of the time window in order to minimize the excess time spent before the time windows of all possible successors open if the vehicle continues to a successor as quickly as possible. The two remaining rules use the same principles to adjust the closing of the time window. With respect to capacity, an arc (i, j) can obviously be removed if $d_i + d_j > q$.

7.2 Subproblem strategies

A well known strategy for accelerating column generation is to return many negative marginal cost columns to the master problem. Even though in principle only one needs to be returned, several can be if they are available. Computational tests conducted by Kohl (1995); Larsen (1999) confirm the benefits of this approach.

7.3 Master problem strategies

Along with the novel perspectives on the subproblem solution described in 5, master problem acceleration strategies have been key to the evolution of VRPTW approaches over the last few years. One approach is to accelerate the solution at the root node of the branch-and-bound tree by using a local search method to generate a set of initial columns. This helps the column generation process get a fast increase in the quality of the dual variables. It has been implemented by numerous researchers and has finally been discussed in the literature by Danna and Le Pape (2005). The authors use a local search method based on the savings algorithm incorporating time windows which produces a set of routes better than the trivial depot-customer-depot ones. Furthermore, local search is used along with a MIP solver throughout the branch-and-price process to generate good integer solutions fast. Two different heuristics, a local search method based on large neighborhood search and a guided tabu search, were tested and proved beneficial, especially on Solomon's R1 and RC1 problem classes.

Two new approaches have been suggested by Larsen (2004, 1999). First, during the execution of the branch-and-price a large number of columns are generated and many of these only participate in a few computations and will not be used afterwards. If kept, each column will increase computing time when solving the relaxed set partitioning problem and when adjusting the upper bounds on variables due to branching decisions. Therefore Larsen (2004) suggests to keep track of how long a column is part of a basis. If it does not participate in a basis for a given number of branch-and-bound nodes it is removed from the model. This was also suggested by Desaulniers, Desrosiers and Solomon (2002) where it was also noted that a certain number of nonbasic columns should remain in the problem. Larsen (2004) reports that deleting columns that have not been part of the basis for the last 20 branch-and-bound nodes outperforms the code without column deletion by a factor of 2.5 aggregated over 27 instances.

The second acceleration approach is to stop the algorithm for the SPPTWCC before it completes. Computations can be stopped as soon as at least one route with negative cost has been generated. This approach is denoted "forced early stop" in Larsen (1999) and results in dramatic running time reductions, especially for problems with large time windows. For these, the values of the dual variables at the beginning of the procedure will however be of poor quality. Only when the subproblem proves optimality it cannot be stopped prematurely.

7.4 Cutting planes

The barebone column generation methodology for solving the VRPTW is part of the popular approach for solving difficult integer programming problems by relaxing the integrality constraints of the original problem. Typically, the optimal solution to the relaxed problem is not feasible for the original problem and branch-and-bound is used in order to get integer solutions.

Cutting planes has been proposed to improve the polyhedral description of the relaxed problem in order to get an integer solution or at least narrow the integrality gap. Kohl, Desrosiers, Madsen, Solomon and Soumis (1999) suggested three cuts in order to tighten the LP formulation of the VRPTW problem. As these cuts are only introduced at the root node, this is not a branch-and-cut approach, where cuts can be introduced at any node of the search tree.

The method is based on subtour elimination constraints and comb inequalities transferred from the TSP, and 2-path cuts. To detect subtour elimination constraints, a separation algorithm by Crowder and Padberg (1980) was implemented. With respect to the comb inequalities, only combs with 3 teeth and 2 nodes were detected. The separation algorithm was a primitive enumeration scheme. Neither of these constraints had a large impact on tightening the bound.

A new idea introduced by Kohl, Desrosiers, Madsen, Solomon and Soumis (1999) was the inclusion of 2-path cuts. The basis of this set of cuts is the subtour elimination inequality in the strong form: $x(S) \geq k(S), \forall S \subseteq C$, where $x(S)$ is the flow leaving the set S, and $k(S)$ is the minimum number of vehicles needed to service the customers in S. Determining $k(S)$ is not an easy task, but using the triangle inequality on the travel times we have that $S_1 \subset S_2 \Rightarrow k(S_1) \leq k(S_2)$. Sets S that satisfy $x(S) < 2$ and $k(S) > 1$ must now be found. As $k(S)$ is an integer, $k(S) > 1$ implies $k(S) \geq 2$. So we need to identify sets S that require at least two vehicles to be serviced, but are currently serviced by less than two.

For a set S, two checks have to be performed: 1) $k(S) > 1$ and 2) can the customers be serviced by a single vehicle? The first check is easy, but the second requires the solution of the TSPTW *feasibility* problem. Since this problem is NP-hard the separation algorithm can only be applied to small sets. This is done heuristically using a simple greedy algorithm based on Laporte, Nobert and Desrochers (1985).

The 2-path cuts outperformed the branch-and-price method without 2-path cuts. The proportion of the integrality gap closed by the 2-path cuts varies from 100% to 10% in a few cases. Overall 12 new unsolved Solomon instances were closed.

Cook and Rich (1999) extended the above 2-path cut approach to k-path cuts involving the solution of a VRPTW with $(k-1)$ customers as part of the separation algorithm. The authors performed experiments with k up to 6. For larger k, the percentage of the integrality gap that is closed is of course larger, but the separation algorithm requires substantially more time and therefore it is not evident that it is preferable to use k larger than 2.

Recently, Bard, Kontoravdis and Yu (2002) have proposed a branch-and-cut algorithm for the arc formulation of the VRPTW. This development parallels the initial uses of this technique for the VRP (Naddef and Rinaldi, 2002). Based on the results obtained by Mak (2001), a new arc formulation of the VRPTW is presented in Kallehauge and Boland (2004). In this formulation the time and capacity restrictions are modeled using infeasible path elimination constraints (IPECS). This new class of inequalities can be viewed as a strengthening of the IPECS described in Ascheuer, Fischetti and Grötschel (2000), Ascheuer, Fischetti and Grötschel (2001); Bard, Kontoravdis and Yu (2002) and can also be incorporated at the master problem level in the path formulation considered in this chapter.

Another line of research involves valid inequalities derived from the precedence relationships established by the time windows. That is, if a set of customers is served by the same vehicle, the associated time windows create a precedence structure among the corresponding nodes (Ascheuer, Fischetti and Grötschel, 2001). In Kallehauge and Boland (2004), two classes of valid inequalities for the precedence-constrained asymmetric traveling salesman polytope (Balas, Fischetti and Pulleyblank, 1995) are transferred to the VRPTW.

8. Generalizations of the VRPTW model

The methods considered in this chapter can be generalized and applied to a number of related problems as discussed by Desrosiers, Dumas, Solomon and Soumis (1995). Here we will concentrate on routing generalizations and show how a number of more complex routing problems can be modeled based on the framework introduced in the previous sections.

8.1 Non-identical vehicles

In the general case vehicles may differ with respect to travel time, travel costs, capacity and possibly other characteristics. We define a class of vehicles as a set of identical vehicles. There may be a cost associated with the vehicles of a particular class, and there may be bounds on their availability as well. These bounds are modeled in to the master problem as supplementary constraints.

The subproblem must be solved separately for each class of vehicles. The marginal costs of the arcs originating at the depot of the subproblem for a particular vehicle class must be modified by the simplex multiplier of the constraints on the availability of this class in the master problem. One can chose to solve one or more of the subproblems between each master iteration. The LP optimality criterion is that no subproblem generates columns with negative reduced costs. It is likely to be efficient to branch on the number of vehicles of a particular class if this number is fractional.

A special case occurs if vehicles do not differ with respect to traveling time, travel cost and time windows, but only have different capacities and possible availability and fixed costs. This problem is clearly solvable as described above, but it can also be transformed into the identical vehicle problem described earlier in this chapter. The advantage of this transformation is that only one subproblem must be solved at each iteration. To illustrate how the transformation works consider a problem with two classes of vehicles, with vehicle capacities q_1 and q_2 respectively, where $q_1 < q_2$. The fixed costs of using the vehicles are c_1 and c_2, respectively. Two extra nodes are inserted in parallel between the depot and the customers and any path must go through exactly one of these nodes. The two arcs from the depot to the new nodes are priced c_1 and c_2, respectively. If node 1 is chosen, the capacity is reduced by $q_2 - q_1$ since the resource window of node 1 starts at this quantity. Since the resource window of the depot is $[0, q_2]$, a path going through node 1 cannot service customers with accumulated demand of more than $q_2 - (q_2 - q_1) = q_1$. If there are bounds on the availability of the vehicles, these are inserted in the master problem and the simplex multipliers modify the cost of the two new arcs between the depot and the new nodes.

8.2 Multiple depots

If the vehicles are based at different depots, one subproblem must be solved for each depot. Constraints on the availability of vehicles at a particular depot are kept in the master problem, and the associated simplex multiplier modifies the cost of arcs originating at the depot. This is equivalent to the general non-identical vehicle case discussed above.

One may assume that the vehicles are allowed to finish their routes at a depot different from the one the vehicles started, but that the number of vehicles starting and ending at any depot remains constant. In this particular case it is sufficient to solve one subproblem. One extra node per depot is created “before” the customers and one ”after” the customers. For each depot there will be a constraint r in the master problem requiring the number of vehicles housed at that depot be kept constant. The right hand side will be zero, and the left hand side coefficient (r, p) will be 1 if route p starts at the depot associated with constraint r and ends at another depot, -1 if the route starts at another depot and ends at the depot associated with constraint r, and zero otherwise. The corresponding simplex multipliers modify the cost of arcs originating at the depot (with opposite sign). It is also easy to introduce different fixed costs associated with the vehicles housed at the depots.

8.3 Multiple or soft time windows

Customers may have several (disjoint) time intervals in which they can be serviced. A vehicle arriving between two time windows must wait until the beginning of the next time window. This doesn't truly complicate the problem since the usual dominance criterion in the subproblem remains valid. A vehicle arriving at a particular node at time t_1 can do everything a vehicle arriving at time t_2 can, provided that $t_1 < t_2$.

If there exist a cost $c(s_i)$ dependent on the time s_i service at customer i begins, the time window is said to be soft. If the cost is non-decreasing with increasing time this is not problematic, since the dominance criteria remain valid. The most general case where $c(s_i)$ is a general function is not efficiently solvable. Ioachim, Gélinas, Desrosiers and Soumis (1998) present an algorithm for the linear case.

9. Computational experiments

Almost from the first computational experiments, a set of problems became the test-bed for both heuristic and exact investigations of the VRPTW. Solomon (1987) proposed a set of 164 instances that have remained the leading test set ever since. For the researchers working on heuristic algorithms for the VRPTW a need for bigger problems made Homberger and Gehring (1999) propose a series of extended Solomon problems. These larger problems have as many as 1000 customers and several have been solved by exact methods.

9.1 The Solomon instances

The test sets reflect several structural factors in vehicle routing and scheduling such as geographical data, number of customers serviced by a single vehicle and the characteristics of the time windows (e.g., tightness, positioning and the fraction of time-constrained customers in the instances). Customers are distributed within a $[0, 100]^2$ square.

The instances are divided into 6 groups (test-sets) denoted R1, R2, C1, C2, RC1 and RC2. Each of the test sets contain between 8 and 12 instances. In R1 and R2 the geographical data is randomly generated by a random uniform distribution. In the test sets C1 and C2 the customers are placed in clusters, and finally in the RC1 and RC2 test-sets some customers are placed in clusters while others are placed randomly. In the test sets R1, C1 and RC1 the scheduling horizon is short permitting approximately 5 to 10 customers to be serviced on each route. The R2, C2 and RC2 problems have a long scheduling horizon allowing routes with more than 30 customers to be feasible. This makes the problems very hard to solve exactly and they have not been used until recently to test exact methods. The time windows for the test sets C1 and C2 are generated to permit good, maybe even optimal, cluster-by-cluster solutions. For each class of problems the geographical position of the customers is the same in all instances whereas the time windows are changed.

Each instance has 100 customers, but by considering only the first 25 or 50 customers, smaller instances can easily be generated. It should be noted that for the RC-sets this results in the customers being clustered since the clustered customers appear at the beginning of the file. Travel time between two customers is usually assumed to be equal to the travel distance plus the service time at the predecessor customer.

9.2 Computational results

This section reviews the results obtained by the best exact algorithms for the VRPTW. All are based on the column generation approach. The tables 3.1 through 3.6 present the solutions for the six different sets of the Solomon instances that have been solved to optimality. Column K indicates the number of vehicles used in the optimal solution while the column "Authors" give reference to the first publication(s) of the optimal solution for the problem: Kohl, Desrosiers, Madsen, Solomon and Soumis (1999) (KDMSS), Larsen (1999) (L), Kallehauge, Larsen and Madsen (2000) (KLM), Cook and Rich (1999) (CR), Irnich and Villeneuve (2005) (IV), Chabrier (2005) (C), and Danna and Le Pape (2005) (DLP). It should be noted that Desrochers, Desrosiers and Solomon (1992) prior to Kohl, Desrosiers, Madsen, Solomon and Soumis (1999) solved 50 of the 87 Solomon problems with narrow time windows, but with different travel times. Whereas all the above mentioned papers compute the travel times using one decimal point precision and truncation, time and cost is computed differently in Desrochers, Desrosiers and Solomon (1992). Furthermore, solutions to all C1 instances were reported for the first time by Kohl and Madsen (1997), who used a Lagrangian relaxation approach.

As discussed in Cordeau, Desaulniers, Desrosiers, Solomon, and Soumis (2002), the optimal algorithm of Kohl, Desrosiers, Madsen, Solomon and Soumis (1999) solved 69 of the 87 Solomon benchmark short horizon problems to optimality. Eleven additional problems were solved by

Table 3.1. Optimal solutions for the R1 instances.

Problem	K	Dist.	Authors	Problem	K	Dist.	Authors
R101.25	8	617.1	KDMSS	R107.25	4	424.3	KDMSS
R101.50	12	1044	KDMSS	R107.50	7	711.1	KDMSS
R101.100	20	1637.7	KDMSS	R107.100	11	1064.6	CR+KLM
R102.25	7	547.1	KDMSS	R108.25	4	397.3	KDMSS
R102.50	11	909	KDMSS	R108.50	6	617.7	CR+KLM
R102.100	18	1466.6	KDMSS	R108.100			
R103.25	5	454.6	KDMSS	R109.25	5	441.3	KDMSS
R103.50	9	772.9	KDMSS	R109.50	8	786.8	KDMSS
R103.100	14	1208.7	CR+L	R109.100	13	1146.9	CR+KLM
R104.25	4	416.9	KDMSS	R110.25	5	444.1	KDMSS
R104.50	6	625.4	KDMSS	R110.50	7	697	KDMSS
R104.100	11	971.5	IV	R110.100	12	1068	CR+KLM
R105.25	6	530.5	KDMSS	R111.25	4	428.8	KDMSS
R105.50	9	899.3	KDMSS	R111.50	7	707.2	CR+KLM
R105.100	15	1355.3	KDMSS	R111.100	12	1048.7	CR+KLM
R106.25	5	465.4	KDMSS	R112.25	4	393	KDMSS
R106.50	8	793	KDMSS	R112.50	6	630.2	CR+KLM
R106.100	13	1234.6	CR+KLM	R112.100			

Larsen (1999); Cook and Rich (1999); Kallehauge, Larsen and Madsen (2000). Recently, Irnich and Villeneuve (2005) were successful in closing three additional instances. Four 100-customer instances are still open.

As also reported in Cordeau, Desaulniers, Desrosiers, Solomon, and Soumis (2002); Larsen (1999); Cook and Rich (1999); Kallehauge, Larsen and Madsen (2000) also provided exact solutions to 42 of the 81 Solomon long horizon problems. Since then, Irnich and Villeneuve (2005); Chabrier (2005); Danna and Le Pape (2005) have solved an additional 21 instances, leaving 18 problems still unsolved.

10. Conclusions

In this chapter we have highlighted the noteworthy developments for optimal column generation approaches to the VRPTW. To date, such methods incorporating branching and cutting on solutions obtained through Dantzig-Wolfe decomposition are the best performing algorithms. Valid inequalities have proved an invaluable tool in strengthening the LP relaxation for this class of problems.

Table 3.2. Optimal solutions for the C1 instances

Problem	*K*	*Dist.*	*Authors*	*Problem*	*K*	*Dist.*	*Authors*
C101.25	3	191.3	KDMSS	C106.25	3	191.3	KDMSS
C101.50	5	362.4	KDMSS	C106.50	5	362.4	KDMSS
C101.100	10	827.3	KDMSS	C106.100	10	827.3	KDMSS
C102.25	3	190.3	KDMSS	C107.25	3	191.3	KDMSS
C102.50	5	361.4	KDMSS	C107.50	5	362.4	KDMSS
C102.100	10	827.3	KDMSS	C107.100	10	827.3	KDMSS
C103.25	3	190.3	KDMSS	C108.25	3	191.3	KDMSS
C103.50	5	361.4	KDMSS	C108.50	5	362.4	KDMSS
C103.100	10	826.3	KDMSS	C108.100	10	827.3	KDMSS
C104.25	3	186.9	KDMSS	C109.25	3	191.3	KDMSS
C104.50	5	358	KDMSS	C109.50	5	362.4	KDMSS
C104.100	10	822.9	KDMSS	C109.100	10	827.3	KDMSS
C105.25	3	191.3	KDMSS				
C105.50	5	362.4	KDMSS				
C105.100	10	827.3	KDMSS				

Table 3.3. Optimal solutions for the RC1 instances.

Problem	*K*	Dist.	Authors	Problem	*K*	Dist.	Authors
RC101.25	4	461.1	KDMSS	RC105.25	4	411.3	KDMSS
RC101.50	8	944	KDMSS	RC105.50	8	855.3	KDMSS
RC101.100	15	1619.8	KDMSS	RC105.100	15	1513.7	KDMSS
RC102.25	3	351.8	KDMSS	RC106.25	3	345.5	KDMSS
RC102.50	7	822.5	KDMSS	RC106.50	6	723.2	KDMSS
RC102.100	14	1457.4	CR+KLM	RC106.100			
RC103.25	3	332.8	KDMSS	RC107.25	3	298.3	KDMSS
RC103.50	6	710.9	KDMSS	RC107.50	6	642.7	KDMSS
RC103.100	11	1258	CR+KLM	RC107.100	12	1207.8	IV
RC104.25	3	306.6	KDMSS	RC108.25	3	294.5	KDMSS
RC104.50	5	545.8	KDMSS	RC108.50	6	598.1	KDMSS
RC104.100				RC108.100	11	1114.2	IV

Recent advances have stemmed from work on parallel implementations of the overall approach, acceleration strategies, primarily at the master problem level, and the subproblem. Solving the subproblem as a ESPPTWCC or a SPPTWCC-k-cyc has shown to be very beneficial.

Table 3.4. Optimal solutions for the R2 instances.

Problem	K	Dist.	Authors	Problem	K	Dist.	Authors
R201.25	4	463.3	CR+KLM	R207.25	3	361.6	KLM
R201.50	6	791.9	CR+KLM	R207.50			
R201.100	8	1143.2	KLM	R207.100			
R202.25	4	410.5	CR+KLM	R208.25	1	328.2	IV+C
R202.50	5	698.5	CR+KLM	R208.50			
R202.100				R208.100			
R203.25	3	391.4	CR+KLM	R209.25	2	370.7	KLM
R203.50	5	605.3	IV+C	R209.50	4	600.6	IV+C
R203.100				R209.100			
R204.25	2	355	IV+C	R210.25	3	404.6	CR+KLM
R204.50	2	506.4	IV	R210.50	4	645.6	IV+C
R204.100				R210.100			
R205.25	3	393	CR+KLM	R211.25	2	350.9	KLM
R205.50	4	690.1	IV+C	R211.50	3	535.5	IV+DLP
R205.100				R211.100			
R206.25	3	374.4	CR+KLM				
R206.50	4	632.4	IV+C				
R206.100							

Table 3.5. Optimal solutions for the C2 instances.

Problem	K	*Dist.*	*Authors*	*Problem*	K	*Dist.*	*Authors*
C201.25	2	214.7	CR+L	C205.25	2	214.7	CR+L
C201.50	3	360.2	CR+L	C205.50	3	359.8	CR+KLM
C201.100	3	589.1	CR+KLM	C205.100	3	586.4	CR+KLM
C202.25	2	214.7	CR+L	C206.25	2	214.7	CR+L
C202.50	3	360.2	CR+KLM	C206.50	3	359.8	CR+KLM
C202.100	3	589.1	CR+KLM	C206.100	3	586	CR+KLM
C203.25	2	214.7	CR+L	C207.25	2	214.5	CR+L
C203.50	3	359.8	CR+KLM	C207.50	3	359.6	CR+KLM
C203.100	3	588.7	KLM	C207.100	3	585.8	CR+KLM
C204.25	1	213.1	CR+KLM	C208.25	2	214.5	CR+L
C204.50	2	350.1	KLM	C208.50	2	350.5	CR+KLM
C204.100	3	588.1	IV	C208.100	3	585.8	KLM

Table 3.6. Optimal solutions for the RC2 instances.

Problem	K	Dist.	Authors	Problem	K	Dist.	Authors
RC201.25	3	360.2	CR+L	RC205.25	3	338	L+KLM
RC201.50	5	684.8	L+KLM	RC205.50	5	630.2	IV+C
RC201.100	9	1261.8	KLM	RC205.100	7	1154	IV+C
RC202.25	3	338	CR+KLM	RC206.25	3	324	KLM
RC202.50	5	613.6	IV+C	RC206.50	5	610	IV+C
RC202.100	8	1092.3	IV+C	RC206.100			
RC203.25	2	326.9	IV+C	RC207.25	3	298.3	KLM
RC203.50	4	490.122	IV+C	RC207.50	4	558.6	C
RC203.100				RC207.100			
RC204.25	3	299.7	C	RC208.25	2	269.1	C
RC204.50	3	444.2	DLP	RC208.50			
RC204.100				RC208.100			

Nevertheless, 25% of Solomon's problems are still unsolved. Additional research in each of these areas should lead to further advances. We expect that the further study of polyhedral structures, paralellism, acceleration strategies, and the subproblem will constitute the backbone of research in this area for the next several years. Master problem acceleration methods relying on local search heuristics is just beginning. Other strategies may consider the principle of *stabilization* for column generation discussed in du Merle, Villeneuve, Desrosiers and Hansen (1999) for the VRPTW. Speedup factors of 1 to 10 were achieved by using stabilized column generation on the airline crew pairing problem which closely related to the VRPTW.

Decomposition algorithms are also easily adaptable to other settings. This is because they comprise modules, such as dynamic programming, that can handle a variety of objectives. Lateness, for one, is becoming an increasingly important benchmark in today's supply chains that emphasize on time deliveries. Moreover, they can be run as optimization-based heuristics by means of early stopping criteria.

We hope that this chapter has shed sufficient light on current developments to lead to exciting further research.

Acknowledgments

The research of Marius M. Solomon was partially supported by the Patrick F. and Helen C. Walsh Research Professorship.

References

Ascheuer, N., Fischetti, M., and Grötschel, M. (2000). Polyhedral study of the asymmetric traveling salesman problem with time windows. *Networks* 36:69–79.

Ascheuer, N., Fischetti, M., and Grötschel, M. (2001). Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming* 90:475–506.

Balas, E., Fischetti, M., and Pulleyblank, W.R. (1995). The precedence-constrained asymmetric travelling salesman polytope. *Mathematical Programming* 68:241–265.

Bard, J., Kontoravdis, G., and Yu, G. (2002). A Branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science* 36:250–269.

Chabrier, A. (2005). Vehicle routing problem with elementary shortest path based column generation. Forthcoming in: *Computers and Operations Research.*

Chabrier, A., Danna, E., and Claude Le Pape. (2002). Coopération entre génération de colonnes avec tournées sans cycle et recherche locale appliquée au routage de véhicules. In: *Huitièmes Journées Nationales sur la Résolution de Problèmes NP-Complets* (JNPC 2002).

Cook, W. and Rich, J.L. (1999). A parallel cutting-plane algorithm for the vehicle routing problem with time windows. *Technical report* TR99-04. Department of Computational and Applied Mathematics, Rice University.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., and Soumis, F. (2002). The VRP with time windows. In: *The vehicle routing problem* (P. Toth and D. Vigo, eds.) , pp. 157–193, SIAM Monographs on Discrete Mathematics and its Applications.

Crowder, H. and Padberg, M. (1980). Solving large-scale symmetric travelling salesman problems to optimality. *Management Science* 26:495–509.

Danna, E. and Le Pape, C. (2005). Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows. This volume.

Desaulniers, G., Desrosiers, J., and Solomon, M.M. (2002). Accelerating Strategies in column generation methods for vehicle routing and crew scheduling problems. In: *Essays and Surveys in Metaheuristics* (C. Ribeiro and P. Hansen, eds.), pp. 309–324, Kluwer Academic Publishers.

Desrosiers, M., Desrosiers, J., and Solomon, M.M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* 40:342–354.

Desrochers, M. and Soumis, F. (1988). A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFORMS* 26:191–212.

Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F. (1995). Time constrained routing and scheduling. In: *Handbooks in Operations Research and Management Sciences* (M. Ball, T. Magnanti, C. Monma and G. Nemhauser, eds.), vol. 8, Network Routing, pp. 35–139, North-Holland, Amsterdam.

Desrosiers, J., Sauvé, M., and Soumis, F. (1988). Lagrangean relaxation methods for solving the minimum fleet size multiple travelling-salesman problem with time windows. *Management Science* 34:1005–1022.

Dror, M. (1994). Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research* 42:977–978.

Dumitrescu, I., and Boland, N. (2003). Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks* 42:135–153.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* 44:216–229.

Gélinas, S., Desrochers, M., Desrosiers, J., and Solomon, M.M. (1995). A new branching strategy for time constrained routing problems with application to backhauling. *Annals of Operations Research* 61:91–109.

Halse, K. (1992). Modeling and solving complex vehicle routing problems. *Ph.D Thesis*, Technical University of Denmark.

Homberger, J. and Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFORMS* 37:297–318.

Ioachim, I., Gélinas, S., Desrosiers, J., and Soumis F. (1998). A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks*, 31:193–204.

Irnich, S. and Villeneuve, D. (2005). The shortest path problem with k-cycle elimination ($k \geq 3$): Improving a branch-and-price algorithm for the VRPTW. Forthcoming in: *INFORMS Journal of Computing.*

Jörnsten, K., Madsen, O.B.G., and Sørensen, B. (1986). Exact solution of the vehicle routing and scheduling problem with time windows by variable splitting, *Research Report* 5/1986, Institute of Mathematical Statistics and Operations Research, Technical University of Denmark.

Kallehauge, B. (2000). Lagrangian duality and non-differentiable optimization—Applied on vehicle routing [in Danish]. *Master's Thesis*, Technical University of Denmark.

Kallehauge, B. and Boland, N. (2004). Infeasible path inequalities for the vehicle routing problem with time windows. *Technical Report*, Centre for Traffic and Transport, Technical University of Denmark.

Kallehauge, B., Larsen, J., and Madsen, O.B.G. (2000). Lagrangean duality and non-differentiable optimization applied on routing with time windows—Experimental results. *Technical Report* IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark.

Kohl, N. (1995). Exact methods for time constrained routing and related scheduling problems. *Ph.D Thesis*, Technical University of Denmark.

Kohl, N., Desrosiers, J., Madsen, O.B.G., Solomon, M.M., and Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science* 33:101–116.

Kohl, N. and Madsen, O.B.G. (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangean relaxation. *Operations Research* 45:395–403.

Kolen, A.W.J., Rinnooy Kan, A.H.G., and Trienekens, H.W.J.M. (1987). Vehicle routing with time windows. *Operations Research* 35:266–273.

Kontoravdis, G. and Bard, J.F. (1995). A GRASP for the vehicle routing problem with time windows. *ORSA Journal on Computing* 7:10–23.

Laporte, G., Nobert, Y., and Desrochers, M. (1985). Optimal routing under capacity and distance restrictions. *Operations Research* 33:1050–1073.

Larsen, J. (1999). Parallelization of the vehicle routing problem with time windows. *Ph.D Thesis*, Technical University of Denmark.

Larsen, J. (2004). Refinements of the column generation process for the vehicle routing problem with time windows. *Journal of Systems Science and Systems Engineering* 13(3):326–341.

Mak, V.H. (2001). On the asymmetric travelling salesman problem with replenishment arcs. *Ph.D Thesis*, Department of Mathematics and Statistics, The University of Melbourne.

du Merle, O., Villeneuve, D., Desrosiers, J., and Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics* 199:229–237.

Naddef, D., and Rinaldi, G. (2002). Branch-and-cut algorithms for the capacitated VRP. In: *The Vehicle Routing Problem* (P. Toth and D. Vigo, eds.), SIAM Monographs on Discrete Mathematics and Applications, pp. 53–84, SIAM, Philadelphia.

Rousseau, L.-M., Gendreau, M., and Pesant, G. (2004). Solving VRPTWs with constraint programming based column generation. *Annals of Operations Research* 130:199–216.

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