Available online at www.sciencedirect.com

ScienceDirect

Computers & Operations Research 35 (2008) 2307–2330

computers & operations research

www.elsevier.com/locate/cor

Formulations and exact algorithms for the vehicle routing problem with time windows

Brian Kallehauge

Centre for Traffic and Transport, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

Available online 18 January 2007

Abstract

In this paper we review the exact algorithms proposed in the last three decades for the solution of the vehicle routing problem with time windows (VRPTW). The exact algorithms for the VRPTW are in many aspects inherited from work on the traveling salesman problem (TSP). In recognition of this fact this paper is structured relative to four seminal papers concerning the formulation and exact solution of the TSP, i.e. the arc formulation, the arc-node formulation, the spanning tree formulation, and the path formulation. We give a detailed analysis of the formulations of the VRPTW and a review of the literature related to the different formulations. There are two main lines of development in relation to the exact algorithms for the VRPTW. One is concerned with the general decomposition approach and the solution to certain dual problems associated with the VRPTW. Another more recent direction is concerned with the analysis of the polyhedral structure of the VRPTW. We conclude by examining possible future lines of research in the area of the VRPTW.

Keywords: Vehicle routing; Time windows; Exact algorithms

1. Introduction

In 1959, a paper by Dantzig and Ramser [1] appeared in the journal Management Science concerning the routing of a fleet of gasoline delivery trucks between a bulk terminal and a number of service stations supplied by the terminal. The distance between any two locations is given and a demand for a certain product is specified for the service stations. The problem is to assign service stations to trucks such that all station demands are satisfied and total mileage covered by the fleet is minimized. The authors imposed the additional conditions that each service station is visited by exactly one truck and that the total demand of the stations supplied by a certain truck does not exceed the capacity of the truck. The problem formulated in the paper by Dantzig and Ramser [1] was given the name ‘truck dispatching problem’. I do not know who coined the name ‘vehicle routing problem’ (VRP) for Dantzig and Ramser’s problem but it caught on in the literature and is the title of the most recent book on the problem, and some of its main variants, edited by Toth and Vigo [2]. In this book, Toth and Vigo [3] considered branch and bound algorithms for the VRP, Naddef and Rinaldi [4] branch and cut algorithms for the VRP and polyhedral studies, Simchi-Levi [5] set covering based approaches for the VRP, Cordeau et al. [6] the VRP with time windows, Toth and Vigo [7] the VRP with backhauls, and Desaulniers et al. [8] the VRP with pickup and delivery. Furthermore, the book reviews heuristic approaches and issues arising in real-world applications. Now the basic variant of the VRP is often given the name ‘capacitated vehicle routing problem’

E-mail address: bk@ctt.dtu.dk.

0305-0548/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2006.11.006

(CVRP) to distinguish it from other members of the family of VRPs. In this paper we consider the VRP with time windows (VRPTW), where each customer must be visited within a specified time interval, called a time window. We consider the case of hard time windows where a vehicle must wait if it arrives before the customer is ready for service and it is not allowed to arrive late. In the case of soft time windows a violation of the time window constraints is accepted but then a price must be paid.

Dantzig and Ramser [1] described how the VRP may be considered as a generalization of the traveling salesman problem (TSP). They described the generalization of the TSP with multiple salesmen and called this problem the 'clover leaf problem', a name that is the very picture of the problem. If there are m salesmen we will refer to the clover leaf problem as the m-TSP, a less lucid name. If in the m-TSP we impose the condition that specified deliveries be made at every location, excepting the start location, we get Dantzig and Ramser's problem. Obviously the VRP is identical with the m-TSP if the total demand of all locations is less than the capacity of a single vehicle. The standard reference book on the TSP was edited by Lawler et al. [9]. In this book Hoffman and Wolfe [10] describe how the importance of the TSP comes from the fact that it is typical of other problems of its genre: combinatorial optimization.

Dantzig had previously collaborated with Fulkerson and Johnson in developing an exact algorithm to the TSP. The appearance of their paper 'Solution of a large-scale traveling-salesman problem' [11] in the journal Operations Research was according to Hoffman and Wolfe [10] "one of the principal events in the history of combinatorial optimization". In this paper the authors first associated with every tour a vector whose entries are indexed by the roads between the cities. An entry of this vector is 1 whenever the road between a pair of cities is traveled, otherwise it is 0. They also defined the linear equations that ensure all cities are visited exactly once in all representations of tours. These equations are called the degree constraints. Second, they defined a linear objective function that expressed the cost of a tour as the sum of road distances of successive pairs of cities in the tour. The problem is then to minimize the linear objective function such that the degree constraints are satisfied and the solution forms a tour. Third, the authors made a linear programming problem out of this integer programming problem by identifying just enough additional linear constraints on the vectors to assure that the minimum is assumed by some tour. This lead to the introduction of the subtour elimination constraints, which excludes solutions where cities are visited exactly once, but in a set of disconnected subtours. However, the authors pointed out that there are other types of constraints which sometimes must be added in addition to subtour elimination constraints in order to exclude solutions vectors involving fractional entries.

By now the approach of Dantzig et al. [11] is basic in combinatorial optimization. The approach is concerned with identifying linear inequalities or cutting planes describing the polytope defined by the convex hull of the points in the Euclidean space that represents the set of feasible solutions of the combinatorial optimization problem. No full description of the TSP polytope is known and because the TSP belongs to the class of NP-complete combinatorial optimization problems there is no hope for a polynomial-time cutting plane method for the TSP, unless NP=P. However, as Dantzig et al. [11] showed the cutting plane algorithm can still be applied to the TSP by including the TSP polytope in a larger polytope (a relaxation) over which we minimize a linear objective function. In this way the TSP is formulated as a linear program that gives a lower bound for the TSP which can be useful in a branch and bound algorithm. Padberg and Rinaldi [12] refined the integration of the enumeration approach of classical branch and bound algorithms with the polyhedral approach of cutting planes to create the solution technique called branch and cut. This method has been very successful in solving large-scale instances of the TSP and different authors have therefore applied the polyhedral approach to other hard combinatorial optimization problems. Laporte et al. [13] were the first to apply the polyhedral approach to the VRP. Finally, we note that the field of discrete mathematics where combinatorial optimization problems are formulated as linear programs is called polyhedral combinatorics and we refer to the recent work of Schrijver [14] for a detailed treatment of this subject. For a treatment of polyhedral theory we refer to Nemhauser and Wolsey [15].

Now we consider another basic method in combinatorial optimization which is concerned with the characterization of the objective function of the combinatorial optimization problem instead of its polytope. Using relaxation and duality we can determine the optimal objective function value, or at least a good lower bound on it (assuming minimization), without explicitly solving the integer problem. In particular, we are concerned with Lagrangian relaxation and duality. A related technique is Dantzig–Wolfe decomposition, which provides an equivalent bound to the Lagrangian dual bound. In Lagrangian relaxation a set of complicating constraints are dualized into the objective function by associating Lagrangian multipliers with them. This gives us an infinite family of relaxations with respect to the Lagrangian multipliers. For a given set of values of the Lagrangian multipliers the relaxed problem is called the Lagrangian subproblem. The problem of determining the largest lower bound for this family is called the Lagrangian dual problem. A fundamental result in mathematical programming is that the Dantzig–Wolfe (generalized) linear programming problem of finding a convex combination of solutions to the (Lagrangian) subproblem that also satisfy the complicating constraints is dual to the Lagrangian dual problem. The book by Shapiro [16] marked the first appearance of the term Lagrangian relaxation in a textbook. In this book the treatment of duality takes a constructive rather than existential approach to Lagrangian multipliers. Everett [17] was the first to take this constructive point of view of Lagrangian multipliers, which is different from the Karush–Kuhn–Tucker point of view of optimality involving dual variables. For a treatment of Lagrangian duality we refer to Hiriart-Urruty and Lemaréchal [18, Chapter XII]. There exist two classical algorithms for solving the Lagrangian dual problem. The simplest algorithm for the Lagrangian dual problem is the subgradient algorithm. The other classical algorithm is the cutting-plane algorithm (a row-generation algorithm), which in the primal version is the column-generation algorithm. These algorithms are convex minimization algorithms and belong to the field of non-smooth or non-differentiable optimization. For a treatment of non-smooth optimization we also refer to Hiriart-Urruty and Lemaréchal [18]. According to Thienel [19], the combination of branch and bound and column generation was by analogy to branch and cut called branch and price by Savelsbergh. Finally, when both variables and constraints are generated in the nodes of the search tree the procedure is called branch, cut, and price. In the last decade a number of frameworks for implementing branch, cut, and price has appeared, e.g. ABACUS [20], SYMPHONY [21], and BCP [22].

The use of Lagrangian relaxation in combinatorial optimization was in fact also inspired by the successful application of it to the TSP by Held and Karp [23,24]. Lagrangian relaxation translates the problem of minimizing our objective function over a set of linear inequalities to finding the maximum of a concave piecewise affine function. There is a relationship between polyhedral combinatorics and Lagrangian relaxation. It is defined by the set of inequalities describing the convex hull of the incidence vectors of solutions to the (Lagrangian) subproblem. Held and Karp [23] proved using general linear programming theory that the maximization of the bound provided by the 1-tree Lagrangian relaxation of the TSP gives the same bound as the linear programming relaxation of the TSP proposed by Dantzig et al. [11] using the subtour inequalities. In this way the relationship between these two seminal contributions [1,23,24] is established, and thereby also an example of the relationship between polyhedral combinatorics and Lagrangian relaxation. If the complete set of inequalities of the subproblem was known it could be included in the integer programming problem and minimizing our objective function over the set of complicating constraints and the inequalities of the subproblem would give the Lagrangian dual value.

The methods for the VRP with time windows are in many aspects inherited from the work done for the TSP. In recognition of this fact this paper is structured relative to the four seminal papers on the TSP formulations, i.e. the arc formulation, the arc-node formulation, the spanning tree formulation, and the path formulation. We only give a detailed analysis of the formulations in this paper but we do give a full review of the literature related to the different formulations. There are two main lines of development in relation to the exact algorithms. One is concerned with the general decomposition approach and the solution to a certain dual problem associated with the primal VRPTW. Another direction is concerned with the analysis of the polyhedral structure of the VRPTW. The idea of convexity is central to both directions.

In what follows we give the complete list of references for the VRPTW relative to the four seminal papers on the TSP. We give the name of the authors of the papers concerning the TSP followed by a list of the papers on the VRPTW that consider a generalization of the approach for the TSP.

- Arc formulation, Dantzig et al. [11], [25,26].
- Arc-node formulation, Miller et al. [27], [28].
- Spanning tree formulation, Held and Karp [23,24], [29].
- Path formulation, Houck et al. [30], [29,31–46].

Cordeau et al. [6] and Kallehauge et al. [47] also give recent surveys in relation to the VRPTW. The survey of Kallehauge et al. [47] is given in the context of column generation in general and the focus is therefore the path formulation of the VRPTW which has been studied by several authors. These surveys also give a status on the computational success of the state of the art algorithms proposed in the literature.

The main contribution of this paper is that it provides a complete survey of all the formulations that has formed a basis for the exact solution of the VRPTW. The literature is approached from a different angle than the existing surveys and it includes new material on the polyhedral approach and a detailed treatment of the spanning tree formulation which is not available elsewhere. Finally, it contains a review of the recent developments related to the path formulation. The aim is also to provide an introduction to the field for people interested in the exact solution of the VRPTW and related resource constrained combinatorial optimization problems.

This paper is organized as follows. In Section 2 we define the VRPTW as a graph theoretic problem and introduce notation used throughout the paper. We also describe the complexity of the VRPTW and define its polytope. In Section 3 we consider the arc formulation of the VRPTW involving only binary variables associated with arcs of an underlying directed graph. In Section 4 we review the arc-node formulation of the VRPTW where we also associate variables with nodes of the directed graph. Section 5 considers a method to find lower bounds for the VRPTW, with the help of time and capacity constrained shortest spanning trees and Lagrangian relaxation. In Section 6 we consider a method to find lower bounds for the VRPTW, with the help of time and capacity constrained shortest paths and Lagrangian relaxation. Finally, in Section 7 we present some conclusions and discuss future directions of research.

2. Problem definition and notation

In this section we define the VRPTW as a graph theoretic problem and introduce notation used throughout the paper.

Given a set W, we associate the concept of a function $c : W \to \mathbb{R}$ with a vector c in $\mathbb{R}^W$. The components of the vector is denoted by c_w. For any function $c : W \to \mathbb{R}$ and any $F \subseteq W$, we denote

$$c(F) = \sum_{w \in F} c_w.$$

If c is introduced as a “cost function” then c_w is called the cost of w, and for any $F \subseteq W$, $c(F)$ is called the cost of F.

Definition 2.1. A time and capacity constrained digraph $D = (V, A, c, t, a, b, d, q)$ is defined by a node set $V = V_* \cup \{0, n+1\}$ for $V_* = \{1, \ldots, n\}$ the set of customer nodes and 0 and $n+1$, respectively, the start and destination depot node, arc set $A = A_* \cup \delta^+(0) \cup \delta^-(n+1)$ for $A_* = A(V_*)$ the set of arcs spanned by the customer nodes and $\delta^+(0) = \{(0, i) \mid i \in V_*\}$ the set of arcs leaving the start depot node and $\delta^-(n+1) = \{(i, n+1) \mid i \in V_*\}$ the set of arcs entering the destination depot node, costs on arcs $c \in \mathbb{Z}_+^A$ where $c_{ij} \leqslant c_{ik} + c_{kj}$ for $i, j, k \in V$, durations on arcs $t \in \mathbb{Z}_+^A$ where $t_{ij} \leqslant t_{ik} + t_{kj}$ for $i, j, k \in V$, release and due times on nodes $a, b \in \{\mathbb{Z}_+ \cup \{+\infty\}\}^V$ where $a_0 = a_{n+1} = 0$, $b_0 = b_{n+1} = +\infty$, $a_i \geqslant t_{0i}$ and $b_i \geqslant a_i$ for $i \in V_*$ and $b_j \geqslant a_i + t_{ij}$ for $(i, j) \in A_*$, demands on nodes $d \in \mathbb{N}^V$ where $d_0 = d_{n+1} = 0$, and capacity $q \in \mathbb{N}$ where $q \geqslant d_i$ for $i \in V_*$ and $q \geqslant d_i + d_j$ for $(i, j) \in A_*$.

Definition 2.2. The arc durations $t \in \mathbb{Z}_+^A$ are defined by service times on nodes $p \in \mathbb{Z}_+^V$ where $p_0 = p_{n+1} = 0$ and $p_i > 0$ for $i \in V_*$ and travel times on arcs $t' \in \mathbb{Z}_+^A$, i.e. $t_{ij} = p_i + t'_{ij}$ for $(i, j) \in A$.

Remark 2.1. The VRPTW variant, which have received the greatest attention in the scientific literature, has a single depot. All vehicles are based at this depot and must return here after visiting customers. However, it is common to split the single depot into a start and destination depot by making a copy of the original depot node. In this way, the start and destination depot nodes become identical. Because we split the depot the vehicle routes are no longer circuits, but paths from the start depot node to the destination depot node. Definition 2.1 of the graph makes it possible to use same definition for all formulations presented in this paper, particularly for the arc and path formulation of Sections 3 and 6.

Remark 2.2. The main part of the literature on exact methods for the VRPTW is concerned with the variant of the problem where the number of available vehicles is free. However, for VRPs in general, one can often find a limit on the number of available vehicles. Furthermore, the minimization of the number of vehicles is a typical objective for VRPs. We refer to Toth and Vigo [2, Chapter 1] for an extensive treatment of definitions of VRPs.

Remark 2.3. Definition 2.1 is not restrictive. If there exists an arc $(i, j) \in A_*$ such that $b_j < a_i + t_{ij}$ or $q < d_i + d_j$,we can simply remove the arc from A_* since its use would violate time and capacity feasibility. Note that the release time a_i denotes the earliest possible (and the due time b_i the latest possible) starting time for visiting node $i \in V_*$.

Definition 2.3. For any path $P = (v_1, \dots, v_k)$ in D, the arrival times of the set of nodes $V(P)$ of the path is the vector $s \in \mathbb{Z}_+^{V(P)}$ defined by

$$s_{v_1} = a_{v_1},$$

$$s_{v_i} = \max\{s_{v_{i-1}} + t_{v_{i-1}v_i}, a_{v_i}\} \quad \text{for } i = 2, \dots, k,$$

the demand of the path is $d(V(P))$, and the cost of the path is $c(A(P))$, where $A(P)$ is the arcs of the path.

Definition 2.4. We say that a path $P = (v_1, \dots, v_k)$ in D is feasible if

$$s_{v_i} \leqslant b_{v_i} \quad \text{for } i \in V(P) \tag{1}$$

and

$$d(V(P)) \leqslant q. \tag{2}$$

Definition 2.5. We say that a path $P = (v_1, \dots, v_k)$ in D is infeasible if

$$s_{v_i} > b_{v_i} \quad \text{for any } i \in V(P) \tag{3}$$

or

$$d(V(P)) > q. \tag{4}$$

Definition 2.6. An infeasible path $P = (v_1, \dots, v_k)$ in D is said to be minimal infeasible if the subpaths of P induced by depriving $V(P)$ of, respectively, the starting node

$$V(P)\backslash\{v_1\} \tag{5}$$

and the end node

$$V(P)\backslash\{v_k\} \tag{6}$$

are feasible. We denote by $\mathscr{P}_D$ the set of all minimal infeasible paths in D.

Definition 2.7. A route R in D is defined as a feasible path from 0 to $n+1$

$$R = (0, v_2, \dots, v_{k-1}, n+1).$$

We denote by $\mathscr{R}$ the set of all routes in D.

Definition 2.8. A k-route in D is the union of k routes

$$\kappa_k = R_1 \cup R_2 \cup \dots \cup R_k,$$

such that each node $v \in V_*$ belongs to exactly one set $V(R_i)$, $1 \leqslant i \leqslant k$. The cost of a k-route is $c(A(\kappa_k)) = c(\cup A(R_i) \mid i = 1, \dots, k)$.

For any $W \subseteq V_*$, computing the number

$$k(W) = \min\{k \mid \text{a } k\text{-route } \kappa_k \text{ exists in } D(W \cup \{0, n+1\})\}, \tag{7}$$

represents the problem of finding the minimum number of routes required to visit the subset of customer nodes W; we stress the fact that the notation $k(W)$ represents at the same time a number and a problem to solve.

Definition 2.9. A partition of the set of customer nodes V_* induced by a k-route κ_k is called a feasible k-partition. We denote by $K = \{k(V_*), \dots, n\}$ the set of feasible partition sizes.

Definition 2.10. For each $k \in K$ we denote by $\mathscr{R}_k$ the set of k-routes with corresponding partition size k.

Definition 2.11. The set of all k-routes in D for $k \in K$ is denoted by $\mathscr{R}_K = \{\cup \mathscr{R}_k \mid k = k(V_*), \dots, n\}$.

The VRP with time windows is defined as follows. Given a time and capacity constrained digraph D, find a k-route of minimum cost, i.e.

$$\min\{c(A(\kappa_k)) \mid \kappa_k \in \mathscr{R}_K\}. \quad \text{(VRPTW)}$$

2.1. Complexity

If we place the restrictions on the instances of the VRPTW that $a_i = 0$ and $b_i = +\infty$ for every $i \in V_*$ the resulting restricted problem will be identical to the CVRP with a free number of vehicles. Note that the CVRP is usually defined with a fixed number of vehicles Toth and Vigo [2]. If we furthermore place the restriction on the instances of the CVRP with a free number of vehicles that $q \geqslant d(V_*)$ the resulting restricted problem will be identical to the TSP. Furthermore, if we place the restrictions on the instances of the VRPTW that $q \geqslant d(V_*)$, and $c_{ij} = t'_{ij} = 0$ for $(i, j) \in A$ the resulting restricted problem is the non-preemptive single machine scheduling problem (SS1) with release dates a, deadlines b, and processing times p. Garey and Johnson [48] proved that TSP and SS1 are NP-complete in the strong sense. This implies by proof of restriction that VRPTW is NP-complete in the strong sense. Finally, if $a_i = 0$ and $b_i = +\infty$ for every $i \in V_*$ then the number $k(V_*)$ may be determined by solving the bin packing problem (BPP) which Garey and Johnson [48] also proved is NP-complete in the strong sense.

2.2. Polytope

Definition 2.12. With every k-route $\kappa_k \in \mathscr{R}_K$ in D, we associate an incidence vector $x^{\kappa_k} \in \mathbb{R}^A$ defined by

$$x_{ij}^{\kappa_k} = \begin{cases} 1 & \text{if } (i, j) \in A(\kappa_k), \\ 0 & \text{if } (i, j) \notin A(\kappa_k). \end{cases}$$

Definition 2.13. The VRPTW polytope of a time and capacity constrained digraph $D = (V, A, c, t, a, b, d, q)$ is the convex hull of the incidence vectors of the k-routes in $\mathscr{R}_K$:

$$\mathscr{P}_{\text{VRPTW}} = \text{conv}\{\mathrm{x}^{\kappa_k} \in \mathbb{R}^A \mid \kappa_k \in \mathscr{R}_K\}.$$

The VRP with time windows is equivalent to minimizing the function $c^{\mathsf{T}}x$ over the VRPTW polytope. The NP-completeness of the VRPTW also implies that no description in terms of inequalities of the VRPTW polytope may be expected. However, polynomial-time computable lower bounds for the VRPTW can be obtained by including the VRPTW polytope in a larger polytope (a relaxation) over which $c^{\mathsf{T}}x$ can be minimized in polynomial time.

3. Subtour and path inequalities

In this section we consider a formulation of the VRPTW involving only binary variables associated with the arcs in D.

The VRPTW polytope is the set of those $x \in \mathbb{B}^A$ satisfying the degree equations

$$x(\delta^+(i)) = 1 \quad \text{for } i \in V_*, \tag{8}$$

$$x(\delta^-(i)) = 1 \quad \text{for } i \in V_*, \tag{9}$$

the subtour inequalities

$$x(A(W)) \leqslant |W| - 1 \quad \text{for } W \subseteq V_* \text{ with } |W| \geqslant 2, \tag{10}$$

and the path inequalities

$$x(A(P)) \leqslant |A(P)| - 1 \quad \text{for } P \in \mathscr{P}_D. \tag{11}$$

The formulation (8)–(11) of the VRPTW was proposed by Kallehauge et al. [25]. The subtour inequalities were proposed by Dantzig et al. [11] in their seminal paper on the TSP. The idea of using path inequalities to model time window restrictions was presented by Ascheuer et al. [49] in their paper on the ATSPTW.

Laporte et al. [13] generalized the subtour inequalities for the CVRP

$$x(A(W)) \leqslant |W| - \left\lceil \frac{d(W)}{q} \right\rceil \quad \text{for } W \subseteq V_* \text{ with } W \neq \emptyset. \tag{12}$$

Naddef and Rinaldi [4] reviewed capacity inequalities of the CVRP polytope including the rounded capacity inequalities (12). Kohl et al. [35] further generalized the subtour inequalities for the VRPTW

$$x(A(W)) \leqslant |W| - k(W) \quad \text{for } W \subseteq V_* \text{ with } W \neq \emptyset, \tag{13}$$

and denoted them k-path inequalities. If we in the formulation of the VRPTW replace the subtour inequalities (10) with the capacity inequalities (12) then it is sufficient to only include conditions (1) and (3) in Definitions 2.4 and 2.5, respectively. Then we denote by $\mathscr{P}_D^{\mathrm{TW}}$ the set of minimal time infeasible paths of Definition 2.6 and redefine 11 to

$$x(A(P)) \leqslant |A(P)| - 1 \quad \text{for } P \in \mathscr{P}_D^{\mathrm{TW}}. \tag{14}$$

However, further replacing the capacity inequalities (12) with the k-path inequalities (13) is not sufficient to drop (14) in the formulation (8), (9), (12), and (14).

Kallehauge et al. [25] presented a class of strengthened path inequalities S_1 for the VRPTW based on the polyhedral results obtained by Mak [50] in the context of the asymmetric TSP with replenishment arcs. Furthermore, Kallehauge et al. [25] determined the dimension of the VRPTW polytope and proved that the S_1 inequalities are facet defining under certain assumptions. These were the first polyhedral results for the VRPTW. Kallehauge et al. [25] also transferred the precedence constraints of Balas et al. [51] to the VRPTW context. Finally, the authors implemented a branch and cut algorithm that showed promising results and reported a solution to a previously unsolved 50-node test problem of Solomon [52].

Mak and Ernst [26] have also studied a formulation of the VRPTW similar to (8)–(11) and presented five new classes of valid inequalities for the VRPTW. The first four classes are based on the well-known D_k cycle inequalities [53] and the last is a class of path inequalities related to the S_1 class. The authors also proved that the new classes of inequalities are facet defining under certain assumptions.

4. Resource inequalities

Next we introduce a formulation of the VRPTW where we also associate variables with the nodes in D.

The integer solutions of (8)–(11) are exactly the incidence vectors of k-routes, so it gives an integer programming formulation of the VRPTW. The class of subtour inequalities (10) have a cardinality growing exponentially with n. An equivalent class of inequalities with polynomial cardinality was proposed by Tucker in 1960 [27]. He introduced node variables $u \in \mathbb{Z}^{V_*}$ and proposed the inequalities

$$u_i - u_j + p x_{ij} \leqslant p - 1 \quad \text{for } (i, j) \in A_*. \tag{15}$$

where $p \leqslant n$. The node variables u_i play the role of node potentials in an electrical network and the inequalities involving them serve to eliminate routes that do not begin at the start depot node 0 and end at the destination depot node $n+1$. This is already achieved by the subtour inequalities of Dantzig et al. [11]. The u_i can be adjusted so that $u_i = j$ if customer i is the jth customer visited in the route which includes customer i. The node variables u_i therefore represent the accumulated number of visits along a route. The inequalities (15) ensure that no more than p customers are visited in one route. For $p \geqslant n$ we have the standard VRPTW. In this way p is a resource limit on the number of visits in a route and we generally denote (15) resource inequalities.

Kulkarni and Bhave [54] generalized Tucker's inequalities for the CVRP, i.e. introduced a class equivalent to (12) but with polynomial cardinality. If every demand d_i for $i \in V_*$ represents a pick-up at customer i and $y \in \mathbb{Z}^{V_*}$ then

$$y_i - y_j + qx_{ij} \leqslant q - d_j \quad \text{for } (i, j) \in A_*, \tag{16}$$

where $d_i \leqslant y_i \leqslant q$ for $i \in V_*$, are denoted pick-up inequalities. For any route $R = (0, v_2, \ldots, v_{k-1}, n+1)$ where $k \geqslant 3$, the node variables of the route $y_{v_i} \in \mathbb{Z}$, $i = 2, \ldots, k-1$, can be adjusted so that

$$y_{v_i} = \sum_{j=2}^{i} d_{v_j}, \tag{17}$$

where $y_{v_{k-1}} \leqslant q$. In case every demand d_i for $i \in V_*$ represents a delivery to customer i and $y' \in \mathbb{Z}^{V_*}$ then

$$y'_i - y'_j - qx_{ij} \geqslant d_j - q \quad \text{for } (i, j) \in A_*, \tag{18}$$

where $y_i \leqslant q - d_i$ for $i \in V_*$, are denoted delivery inequalities. For any route $R = (0, v_2, \ldots, v_{k-1}, n+1)$ where $k \geqslant 3$, the node variables of the route $y'_{v_i} \in \mathbb{Z}$, $i = 2, \ldots, k-1$, can be adjusted so that

$$y'_{v_i} = d(V(R)) - \sum_{j=2}^{i} d_{v_j}, \tag{19}$$

where $y_{v_2} \leqslant q - d_{v_2}$. In the standard VRPTW it is required that all demands represent a pick-up or alternatively that all demands represent a delivery.

Desrochers and Laporte [55] further generalized Tucker's inequalities for the VRPTW, but in the case of time windows the resource inequalities are not only equivalent to the generalized subtour inequalities (13) of Dantzig et al. [11], but also equivalent to the path inequalities (14). If $s \in \mathbb{Z}^{V_*}$ then

$$s_i - s_j + (b_i + t_{ij} - a_j)x_{ij} \leqslant b_i - a_j \quad \text{for } (i, j) \in A_*, \tag{20}$$

where $a_i \leqslant s_i \leqslant b_i$ for $i \in V_*$, are denoted as the time inequalities.

For any route $R = (0, v_2, \ldots, v_{k-1}, n+1)$ where $k \geqslant 3$, the node variables of the route $s_{v_i} \in \mathbb{Z}$, $i = 2, \ldots, k-1$, can be adjusted so that

$$s_{v_1} = a_{v_1}, \tag{21}$$

$$s_{v_i} = \max\{s_{v_{i-1}} + t_{v_{i-1}v_i}, a_{v_i}\} \quad \text{for } i = 2, \ldots, k, \tag{22}$$

where $s_{v_i} \leqslant b_{v_i}$ for $i = 2, \ldots, k-1$.

The VRPTW polytope is the set of those $x \in \mathbb{B}^A$, $y \in \mathbb{Z}^{V_*}$, and $s \in \mathbb{Z}^{V_*}$ satisfying the degree equations

$$x(\delta^+(i)) = 1 \quad \text{for } i \in V_*, \tag{23}$$

$$x(\delta^-(i)) = 1 \quad \text{for } i \in V_*, \tag{24}$$

the pick-up inequalities

$$y_i - y_j + qx_{ij} \leqslant q - d_j \quad \text{for } (i, j) \in A_*, \tag{25}$$

the time inequalities

$$s_i - s_j + (b_i + t_{ij} - a_j)x_{ij} \leqslant b_i - a_j \quad \text{for } (i, j) \in A_*, \tag{26}$$

and the bounds

$$s_i \leqslant b_i \quad \text{for } i \in V_*, \tag{27}$$

$$s_i \geqslant a_i \quad \text{for } i \in V_*, \tag{28}$$

$$y_i \leqslant q \quad \text{for } i \in V_*, \tag{29}$$

$$y_i \geqslant d_i \quad \text{for } i \in V_*. \tag{30}$$

The formulation (23)–(30) of the VRPTW was proposed by Bard et al. [28]. However, the authors considered the problem (7) of finding the minimum number of routes required to visit the set of customers V_*. This problem is equivalent to minimizing the function $x(\delta^+(0))$ over the VRPTW polytope. Bard et al. [28] proposed the first branch and cut algorithm for the VRPTW based on this formulation. They considered a number of well-known inequalities from the TSP and VRP and proposed two new types of path inequalities taking into account the time windows of the problem. However, from a computational point of view the generalized subtour inequalities were the most effective. A reason for this is that the authors developed efficient heuristics for the separation of subtour inequalities. Furthermore, these path inequalities are quite weak compared to e.g. the S_1 inequalities proposed in the context of the VRPTW by Kallehauge et al. [25]. Brad et al. [28] also used a so-called greedy randomized adaptive search procedure (GRASP) for finding feasible solutions or upper bounds in the search tree. The branch and cut method of Bard et al. [28] showed promising computational results.

Formulating the integer programming model is only the first step when hard optimization problems are solved by branch and cut. The crucial part is the subset one considers of the finite family of defining inequalities of the associated polytope. It is well known that Tucker's inequalities generally provide worse linear programming bounds than families of inequalities with exponential cardinality. In the context of the ATSPTW Ascheuer et al. [56] noted that their model involving only binary variables cannot handle as general objective functions as a model involving node variables. One example could be a makespan type of objective where the total time spent is minimized, i.e. including waiting time. Depending on the application this should therefore be the criterion for considering node variables or not because strengthened path inequalities dominate Tucker's inequalities.

5. Trees

In this section we consider a method to find lower bounds for the VRPTW, with the help of time and capacity constrained shortest spanning trees and Lagrangian relaxation or Dantzig–Wolfe decomposition.

Definition 5.1. A 0-arborescence in D is a subset B of A such that B forms a shortest spanning tree on $V\backslash\{n+1\}$ rooted at node 0 and such that for each node $v \in V_*$ there is a feasible path from 0 to v.

Definition 5.2. A routed arborescence, or just arborescence, in D is a subset T of A such that $T\backslash\delta^-(n+1)$ is a 0-arborescence and such that T contains a subset of arcs entering $n+1$, say $F = T \cap \delta^-(n+1)$, where $|F| = |T \cap \delta^+(0)|$. We denote by $\mathscr{T}$ the collection of all arborescences in D. For any arborescence T in D, the cost is defined by $c(T)$.

The shortest (= minimum cost) arborescence problem with time windows and capacity constraints is defined as follows. Given a time and capacity constrained digraph D, find an arborescence T of minimum cost, i.e.

$$\min\{c(T) \mid T \in \mathscr{T}\}. \quad \text{(SAPTWCC)}$$

Papadimitriou [57] proved the NP-completeness of the capacitated tree problem and therefore by proof of restriction the SAPTWCC is also NP-complete.

Definition 5.3. With every arborescence $T \in \mathscr{T}$ in D, we associate an incidence vector $x^T \in \mathbb{R}^A$ defined by:

$$x_{ij}^T = \begin{cases} 1 & \text{if } (i,j) \in T, \\ 0 & \text{if } (i,j) \notin T. \end{cases}$$

Definition 5.4. The SAPTWCC polytope of a time and capacity constrained digraph D is the convex hull of the incidence vectors of the arborescences in $\mathscr{T}$:

$$\mathscr{P}_{\text{SAPTWCC}} = \text{conv}\{\text{x}^T \in \mathbb{R}^A \mid \text{T} \in \mathscr{T}\}.$$

The shortest arborescence problem with time windows and capacity constraints is equivalent to minimizing the function $c^{\mathsf{T}}x$ over the SAPTWCC polytope.

The SAPTWCC polytope is the set of those $x \in \mathbb{B}^A$ satisfying the indegree equations

$$x(\delta^-(i)) = 1 \quad \text{for } i \in V_*, \tag{31}$$

the cut-set inequalities

$$x(\delta^-(W)) \geqslant 1 \quad \text{for } W \subseteq V_* \text{ with } W \neq \emptyset, \tag{32}$$

the path inequalities

$$x(A(P)) \leqslant |A(P)| - 1 \quad \text{for } P \in \mathscr{P}_D \tag{33}$$

and the flow balance equation

$$x(\delta^+(0)) - x(\delta^-(n+1)) = 0. \tag{34}$$

The SAPTWCC can be solved by considering two separate problems:

- the determination of a shortest 0-arborescence B^* in D, defined by those $x \in \mathbb{B}^A$ satisfying (31), (32), and (33), and
- the determination of a subset F^* of minimum cost arcs entering the destination depot $n+1$, defined by those $x \in \mathbb{B}^A$ satisfying (34).

To see the relationship between the VRPTW and the SAPTWCC we consider a slightly different formulation of the VRPTW equivalent to (8)–(11). The outdegree (8) and indegree (9) equations give us a mean to alter the "inside" form of the subtour inequalities (10). By subtracting (8) from (10) we obtain the "outside" form defined by the set of arcs in D leaving W

$$x(\delta^+(W)) \geqslant 1 \quad \text{for } W \subseteq V_* \text{ with } W \neq \emptyset, \tag{35}$$

and by subtracting (9) from (10) we obtain the outside form defined by the set of arcs in D entering W

$$x(\delta^-(W)) \geqslant 1 \quad \text{for } W \subseteq V_* \text{ with } W \neq \emptyset. \tag{36}$$

Furthermore, the degree equations (8) and (9) together with the subtour inequalities (10) imply that

$$x(\delta^+(0)) - x(\delta^-(n+1)) = 0. \tag{37}$$

The formulation (8)–(11) of the VRPTW is therefore equivalent to

$$x(\delta^+(i)) = 1 \quad \text{for } i \in V_*, \tag{38}$$
$$x(\delta^-(i)) = 1 \quad \text{for } i \in V_*, \tag{39}$$
$$x(\delta^-(W)) \geqslant 1 \quad \text{for } W \subseteq V_* \text{ with } W \neq \emptyset, \tag{40}$$
$$x(A(P)) \leqslant |A(P)| - 1 \quad \text{for } P \in \mathscr{P}_D, \tag{41}$$
$$x(\delta^+(0)) - x(\delta^-(n+1)) = 0. \tag{42}$$

In the formulation (38)–(42) of the VRPTW, the outdegree equations (38) appear as the complicating constraints. If these constraints were not present the VRPTW would reduce to the SAPTWCC. To take advantage of this problem structure we therefore consider the Lagrangian relaxation with respect to the outdegree equations (38). For any $\lambda \in \mathbb{R}^{V_*}$ we consider the Lagrangian function defined by

$$L(\lambda, x) = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} - \sum_{i \in V_*} \lambda_i \left(\sum_{j \in V} x_{ij} - 1 \right) \tag{43}$$
$$= \tilde{c}^{\mathsf{T}} x + \lambda(V_*), \tag{44}$$

where

$$\tilde{c}_{ij} = \begin{cases} c_{ij} - \lambda_i & \text{for } i \in V_*, \quad j \in V, \\ c_{ij} & \text{for } i = 0, \quad j \in V_*. \end{cases} \tag{45}$$

The Lagrangian problem associated with λ is defined by

$$z(\lambda) = \min\{\tilde{c}^{\mathsf{T}} x \mid x \in \mathbb{B}^A \text{satisfies conditions(39)–(42)}\} + \lambda(V_*) \tag{46}$$

where λ is fixed in $\mathbb{R}^{V_*}$. Problem (46) is a SAPTWCC with the cost function given by $\tilde{c} : A \to \mathbb{R}$.

The Lagrangian dual problem is defined by

$$\max\{z(\lambda) \mid \lambda \in \mathbb{R}^{V_*}\}. \tag{47}$$

Definition 5.5. Let the set of arborescences $\mathscr{T}$ be indexed with $k=1,\ldots,|\mathscr{T}|$ so T_k is the kth arborescence and define the cost of the kth arborescence

$$c_k = \sum_{(i,j)\in T_k} c_{ij} x_{ij}^{T_k}$$

and the outdegree of node $i \in V_*$ in the kth arborescence

$$a_{ik} = \sum_{j\in V} x_{ij}^{T_k} \quad \text{for } i = 1, \ldots, n.$$

The Lagrangian problem (46) is then defined by

$$z(\lambda) = \min_{1\leqslant k\leqslant|\mathscr{T}|} \{c_k - a_k^{\mathsf{T}}\lambda\} + \lambda(V_*) \tag{48}$$

and the Lagrangian dual problem is defined by

$$\max_{\lambda\in\mathbb{R}^{V_*}} \left\{ \min_{1\leqslant k\leqslant|\mathscr{T}|} \left\{c_k - a_k^{\mathsf{T}}\lambda\right\} + \lambda(V_*) \right\}. \tag{49}$$

Since $\mathscr{T}$ is finite it allows us to express (49) as the following linear program with many constraints or rows:

$$\begin{aligned} &\max\, \theta + \sum_{i\in V_*} \lambda_i \\ &\text{s.t.} \\ &\theta \leqslant c_k - \sum_{i\in V_*} a_{ik}\lambda_i \quad \text{for } k = 1, \ldots, |\mathscr{T}|, \\ &\lambda_i \in \mathbb{R} \quad \text{for } i \in V_*, \\ &\theta \in \mathbb{R}. \end{aligned} \tag{50}$$

The LP dual of (50) is a linear program with many variables or columns

$$\begin{aligned} &\min \sum_{k=1}^{|\mathscr{T}|} c_k y_k \\ &\text{s.t.} \\ &\sum_{k=1}^{|\mathscr{T}|} a_{ik} y_k = 1 \quad \text{for } i \in V_*, \\ &\sum_{k=1}^{|\mathscr{T}|} y_k = 1, \\ &y_k \geqslant 0 \quad \text{for } k = 1, \ldots, |\mathscr{T}|. \end{aligned} \tag{51}$$

Problem (51) with y_k required to be integral is equivalent to the VRPTW. In case of integrality constraints an optimal solution to (51) must satisfy $y_{k^*}=1$ for some $T_{k^*} \in \mathscr{T}$ and $y_k=0$ for all $T_k \in \mathscr{T} \setminus \{T_{k^*}\}$. Problem (51) is the LP relaxation of the Dantzig–Wolfe decomposition obtained when any solution to the VRPTW is expressed as a non-negative convex combination of arborescences. The relaxation of the VRPTW presented in this section has never been used directly in a branch and bound algorithm. The idea of using shortest spanning trees has been considered in the VRPTW context but only one paper [29] in the literature on the VRPTW has considered this classical approach in vehicle routing.

Held and Karp [23] explored the relationship between the symmetric and asymmetric TSP and shortest spanning trees in undirected and directed graphs, respectively. Consider the symmetric TSP and the complete undirected graph $G=(V, E)$ on n nodes. A 1-tree is a subgraph T of G with nodes $1, 2, \ldots, n$ consisting of a tree on the nodes $2, 3, \ldots, n$ together with two edges incident with node 1. In fact, a 1-tree T in G consists of exactly n arcs, whereas a tree in G consists of $n-1$ arcs. So a 1-tree is a tree with an additional arc added explaining the term 1-tree. A solution to the STSP is precisely a 1-tree in which all nodes have degree 2. The Lagrangian relaxation of the STSP with respect to the degree constraints $x(\delta(v))=2$, $v \in V\setminus\{1\}$, gives a shortest 1-tree Lagrangian problem and the Lagrangian dual problem can be expressed as a pair of linear programs similar to (50) and (51). Held and Karp [23] gave a column generation method and an ascent method for finding the Lagrangian dual value. The column generation method was able to solve the program (51) for most problems with $n=12$ and some problems with $13 \leqslant n \leqslant 20$. On larger problems the convergence was always too slow and the authors noted that this was consistent with the behavior of other column generation techniques at that time referring to the work of Gilmore and Gomory [58]. Held and Karp [23] also described how the approach for the symmetric TSP carry over to the asymmetric case. In this case the authors introduced a type of directed subgraph they called a 1-arborescence, defined as an arborescence (directed tree) rooted at node 1 plus an arc $(v, 1)$ joining some node $v \in V\setminus\{1\}$ to node 1. We remark that the authors notion of a 1-arborescence is slightly different from our r-arborescence, which is defined as an arborescence rooted at node r. Held and Karp [23] also described the extension of the 1-tree approach to the m-STSP in which the degree at node 1 is $2m$. This would later be further generalized to the symmetric CVRP. Held and Karp [23] already added in the proof of their paper that a new method for computing the Lagrangian dual value would be presented in a sequel to the paper. The following paper [24] was a milestone in the subject of Lagrangian relaxation in integer programming. Held and Karp [24] successfully introduced what became known as the subgradient algorithm (a term introduced by Held et al. [59]) and influenced future research dramatically [60]. In 1974 Geoffrion [61] coined the term Lagrangian relaxation to describe the method of Held and Karp [23,24]. Because of the initial use of the subgradient algorithm, Lagrangian relaxation to some extent became synonymous with the subgradient algorithm, which is unfortunate because this algorithm is the simplest algorithm for concave maximization and suffers from several drawbacks [18]. Indeed the method is mainly attractive because it is so simple to implement.

The earliest generalization of the approach by Held and Karp [23] was proposed by Christofides et al. [62] for the SCVRP based on the k-degree center tree (k-DCT) relaxation of the SCVRP. The approach allows for the possibility of single customer routes. Fisher [63] presented a different relaxation of the SCVRP using shortest k-trees. Consider the symmetric CVRP and the complete undirected graph $G=(V, E)$ on n nodes. A k-tree is a subgraph of G consisting of $n-1+k$ edges that span the n nodes. The degree of the depot node 1 is $2k$. As the name suggests this is a generalization of the 1-tree approach of Held and Karp [23]. The author dualized the capacity constraints (12) of Laporte et al. [13] and solved the Lagrangian dual problem using the subgradient algorithm and generation of violated capacity constraints. The Lagrangian dual problem expressed as an LP similar to (50) is exponential in size since it has exponentially many variables as well as constraints corresponding to the number of capacity constraints and k-trees, respectively. Instead of using the subgradient algorithm a cut and column generation algorithm similar to the one proposed by Kallehauge et al. [44] could be used for solving the dual problem.

Fisher [63] also described an extension of his method to the VRPTW. He introduced path inequalities (11) in a formulation of the VRPTW and relaxed these to obtain the same k-tree Lagrangian problem considered for the CVRP. He did not report any computational results. The extension to time windows was developed with K. Jörnsten and O.B.G. Madsen and together Fisher et al. [29] later presented computational results using the k-tree method. However, the shortest tree relaxation of the VRPTW has not been the subject of the same amount of research as the shortest path relaxation and in our view it is at this point an open question whether the formulation described in this section is effective.

We do not consider a formulation of the SAPTWCC with a fixed degree at the root node because we consider the variant of the VRPTW with a free number of vehicles. This is different from the TSP (VRP) where the authors consider rooted trees with a degree constraint at the root node because they consider problems with a fixed number of tours (routes). Toth and Vigo [64] proposed an algorithm for the shortest arborescence problem with capacity constraints (SAPCC), however the extension to time windows has not been considered in the literature. Finally, we refer to Schrijver [14, Section 50.6a] for a complexity survey for shortest spanning trees.

6. Paths

We next consider a method to find lower bounds for the VRPTW, with the help of time and capacity constrained shortest paths and Lagrangian relaxation or Dantzig–Wolfe decomposition.

Definition 6.1. We denote by $D_{0,n+1}$ the time and capacity constrained digraph obtained from D by adding the arc $(0, n+1)$ to the set of arcs A of D. For notational convenience we denote the extended set of arcs by A. The cost $c_{0,n+1}$ and duration $t_{0,n+1}$ on arc $(0, n+1)$ are zero.

The elementary shortest path problem with time windows and capacity constraints is defined as follows. Given a time and capacity constrained digraph $D_{0,n+1}$ and a cost function $\tilde{c} : A \to \mathbb{R}$, find a path from 0 to $n+1$ of minimum cost, i.e.

$$\min\{\tilde{c}(A(R)) \mid R \in \mathscr{R}\}. \quad \text{(ESPPTWCC)}$$

Remark 6.1. The optimal solution to the ESPPTWCC in $D_{0,n+1}$ with respect to the original cost function $c : A \to \mathbb{Z}_+$ is the path defined by the arc $(0, n+1)$ and the cost of the path is therefore 0. However, when arc costs are not required to be non-negative (integers) the solution to the ESPPTWCC may be of negative cost. Indeed, the digraph may contain a negative cycle, i.e. a directed cycle of negative cost.

Dror [65] proved the NP-completeness of the ESPPTWCC. Next we define a relaxation of the ESPPTWCC. A walk in $D_{0,n+1}$ from 0 to $n+1$ is a sequence of m nodes

$$R_{(k)} = (0, v_2, \ldots, v_{m-1}, n+1), \tag{52}$$

where $v_2, \ldots, v_{m-1}$ are not necessarily distinct. If $v_{i+p} \neq v_i$ for $2 \leqslant p \leqslant k$ the walk is called a non-elementary route with no k-cycles. The (non-elementary) shortest path problem with time windows and capacity constraints and no k-cycles is defined as follows. Given a time and capacity constrained digraph $D_{0,n+1}$, find a walk from 0 to $n+1$ of minimum cost containing no k-cycles, i.e.

$$\min\{\tilde{c}(A(R_{(k)})) \mid R_{(k)} \in \mathscr{R}_{(k)}\}. \quad (k\text{-SPPTWCC})$$

Pseudo-polynomial time algorithms exist for the k-SPPTWCC [66,30,32,43]. If no cycle elimination is performed the problem is called the SPPTWCC. In $D_{0,n+1}$ we have that $\mathscr{R}_{(2)} \supseteq \mathscr{R}_{(3)} \supseteq \cdots \supseteq \mathscr{R}_{(n-1)} \supseteq \mathscr{R}$.

Definition 6.2. With every route $R \in \mathscr{R}$ in $D_{0,n+1}$, we associate an incidence vector $x^{\mathrm{T}} \in \mathbb{R}^A$ defined by

$$x_{ij}^R = \begin{cases} 1 & \text{if } (i,j) \in A(R), \\ 0 & \text{if } (i,j) \notin A(R). \end{cases}$$

Definition 6.3. The ESPPTWCC polytope of a time and capacity constrained digraph $D_{0,n+1}$ is the convex hull of the incidence vectors of the routes in $\mathscr{R}$:

$$\mathscr{P}_{\text{ESPPTWCC}} = \text{conv}\{\mathrm{x}^R \in \mathbb{R}^A \mid \mathrm{R} \in \mathscr{R}\}.$$

The elementary shortest path problem with time windows and capacity constraints is equivalent to minimizing the function $\tilde{c}^{\mathrm{T}} x$ over the ESPPTWCC polytope.

The ESPPTWCC polytope is the set of those $x \in \mathbb{B}^A$ satisfying the degree equations

$$x(\delta^+(0)) = 1, \tag{53}$$
$$x(\delta^-(n+1)) = 1, \tag{54}$$

the balance equations

$$x(\delta^-(i))) - x(\delta^+(i)) = 0 \quad \text{for } i \in V_*, \tag{55}$$

the subtour inequalities

$$x(A(W)) \leqslant |W| - 1 \quad \text{for } W \subseteq V_* \text{ with } |W| \geqslant 2, \tag{56}$$

and the path inequalities

$$x(A(P)) \leqslant |A(P)| - 1 \quad \text{for } P \in \mathscr{P}_D. \tag{57}$$

To see the relationship between the VRPTW and the ESPPTWCC we define the VRPTW in a slightly different way. Given a time and capacity constrained digraph $D_{0,n+1}$, find a collection of routes $\{R_k \mid k = 1, \ldots, n\}$ of minimum cost such that each node $v \in V_*$ is visited exactly once, i.e.

$$\begin{aligned} &\min \sum_{1 \leqslant k \leqslant n} c(A(R_k)) \\ &\text{s.t.} \\ &|\cup_{1 \leqslant k \leqslant n} A(R_k) \cap \delta^+(v)| = 1 \quad \text{for } v \in V_*, \\ &R_k \in \mathscr{R} \quad \text{for } k = 1, \ldots, n. \end{aligned} \tag{58}$$

Remark 6.2. Since we have introduced the arc $(0, n+1)$ in $D_{0,n+1}$ the solution to (58) may contain a number of zero-cost routes not visiting any customer nodes and problem (58) is therefore equivalent to the VRPTW.

Definition 6.4. With every collection of routes $\kappa_n = \{R_k \mid k = 1, \ldots, n\}$ in $D_{0,n+1}$, we associate an incidence vector $x^{\kappa_n} \in \mathbb{R}^{A^n}$ defined by

$$x_{ijk}^{\kappa_n} = \begin{cases} 1 & \text{if } (i,j) \in A(R_k), \\ 0 & \text{if } (i,j) \notin A(R_k). \end{cases}$$

The polytope of (58) is the set of those $x \in \mathbb{B}^{A^n}$ where $x_k \in \mathbb{B}^A$ satisfy (53)–(57) for $k = 1, \ldots, n$ and

$$\sum_{1 \leqslant k \leqslant n} \sum_{j \in V} x_{ijk} = 1 \quad \text{for } i \in V_*. \tag{59}$$

The VRP with time windows is therefore equivalent to minimizing the function $\sum_{1 \leqslant k \leqslant n} c^\mathsf{T} x_k$ over the polytope of (58).

In the formulation (53)–(59) of the VRPTW, the constraints (59) appear as coupling constraints, which link the individual variables x_k. If these constraints were not present the VRPTW would reduce to n ESPPTWCC problems, each with the simpler formulation (53)–(57), and thus become considerably simpler. To take advantage of this problem structure we therefore consider the Lagrangian relaxation with respect to the constraints (59). For any $\lambda \in \mathbb{R}^{V_*}$ we consider the Lagrangian function defined by

$$L(\lambda, x) = \sum_{1 \leqslant k \leqslant n} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} - \sum_{i \in V_*} \lambda_i \left(\sum_{1 \leqslant k \leqslant n} \sum_{j \in V} x_{ij} - 1 \right) \tag{60}$$
$$= \sum_{1 \leqslant k \leqslant n} \tilde{c}^\mathsf{T} x_k + \lambda(V_*), \tag{61}$$

where

$$\tilde{c}_{ij} = \begin{cases} c_{ij} - \lambda_i & \text{for } i \in V_*, \quad j \in V, \\ c_{ij} & \text{for } i = 0, \quad j \in V_*. \end{cases} \tag{62}$$

The Lagrangian problem associated with λ is defined by

$$z(\lambda) = \min \left\{ \sum_{1 \leqslant k \leqslant n} \tilde{c}^{\mathsf{T}} x_k \mid x_k \in \mathbb{B}^A \text{ satisfies conditions (53)–(57) for } k = 1, \ldots, n \right\} + \lambda(V_*), \tag{63}$$

where λ is fixed in $\mathbb{R}^{V_*}$. Problem (63) can be solved by considering n ESPPTWCC problems with the cost function $\tilde{c} : A \to \mathbb{R}$. Since the n ESPPTWCC subproblems are identical, one only needs to consider one subproblem and the Lagrangian problem takes the form

$$z(\lambda) = n \min\{\tilde{c}^{\mathsf{T}} x \mid x \in \mathbb{B}^A \text{ satisfies conditions (53)–(57)}\} + \lambda(V_*). \tag{64}$$

The Lagrangian dual problem is defined by

$$\max\{z(\lambda) \mid \lambda \in \mathbb{R}^{V_*}\}. \tag{65}$$

Definition 6.5. Let the set of routes $\mathscr{R}$ be indexed with $k = 1, \ldots, |\mathscr{R}|$ so R_k is the kth route and define the cost of the kth route

$$c_k = \sum_{(i,j) \in A(R_k)} c_{ij} x_{ij}^{R_k}$$

and the number of times node $i \in V_*$ is visited by the kth route

$$a_{ik} = \sum_{j \in V} x_{ij}^{R_k} \quad \text{for } i = 1, \ldots, n.$$

The Lagrangian problem (64) is then defined by

$$z(\lambda) = n \min_{1 \leqslant k \leqslant |\mathscr{R}|} \{c_k - a_k^{\mathsf{T}} \lambda\} + \lambda(V_*) \tag{66}$$

and the Lagrangian dual problem is defined by

$$z_{\mathrm{LD}}(\mathscr{R}) = \max_{\lambda \in \mathbb{R}^{V_*}} \{n \min_{1 \leqslant k \leqslant |\mathscr{R}|} \{c_k - a_k^{\mathsf{T}} \lambda\} + \lambda(V_*)\}. \tag{67}$$

In (67) we have that $z_{\mathrm{LD}}(\mathscr{R}_{(2)}) \leqslant z_{\mathrm{LD}}(\mathscr{R}_{(3)}) \leqslant \cdots \leqslant z_{\mathrm{LD}}(\mathscr{R}_{(n-1)}) \leqslant z_{\mathrm{LD}}(\mathscr{R})$.

Since $\mathscr{R}$ is finite it allows us to express (67) as the following linear program with many constraints or rows:

$$\begin{aligned} &\max n\theta + \sum_{i \in V_*} \lambda_i \\ &\text{s.t.} \\ &\theta \leqslant c_k - \sum_{i \in V_*} a_{ik} \lambda_i \quad \text{for } k = 1, \ldots, |\mathscr{R}|, \\ &\lambda_i \in \mathbb{R} \quad \text{for } i \in V_*, \\ &\theta \in \mathbb{R}. \end{aligned} \tag{68}$$

The LP dual of (68) is a linear program with many variables or columns

$$\begin{aligned}
&\min \sum_{k=1}^{|\mathscr{R}|} c_k y_k \\
&\text{s.t.} \\
&\sum_{k=1}^{|\mathscr{R}|} a_{ik} y_k = 1 \quad \text{for } i \in V_*, \\
&\sum_{k=1}^{|\mathscr{R}|} y_k = n, \\
&y_k \geqslant 0 \quad \text{for } k = 1, \ldots, |\mathscr{R}|.
\end{aligned} \tag{69}$$

Problem (69) with y_k required to be integral is equivalent to the VRPTW. This also holds true if we formulate the linear program with respect to $\mathscr{R}_{(k)}$ for some $2 \leqslant k \leqslant n-1$. Problem (69) is the LP relaxation of the Dantzig–Wolfe decomposition obtained when any solution to the VRPTW is expressed as a non-negative convex combination of routes, however, because the subproblems are identical the convexity constraints have been aggregated [47]. The aggregated formulation is equivalent to the standard set-partitioning formulation of VRPs [5]. There is a benefit in working with the decomposed formulation; it does not suffer from the drawback of symmetry present in the original formulation using the three-index variables x_{ijk} where a given solution can be represented in several ways by permuting the k indexing. The formulation of the VRPTW presented in this section using three-index variables has never been used in a branch and bound algorithm. Problem (69) has been an important construct in the formulation of algorithms for the VRPTW but more recently the dual point of view of (68) has also been considered.

Houck et al. [30] presented a relaxation of the symmetric and asymmetric TSP based on resource-constrained paths. The resource in their path definition is the number of arcs contained in the path and the limit on the consumption of this resource is the number of nodes in the graph n. They called a path containing n arcs an n-path. They called a path elementary if its nodes are all distinct except possibly the first and last node. If we fix a node 1 of the (directed) graph an n-path from node 1 to node 1 is a Hamilton tour if and only if it is elementary. The problem of finding an elementary n-path of minimum cost from node 1 to node 1 is equivalent to the TSP and hence NP-complete. The authors therefore relaxed the condition that the path should be elementary. If $P = (1, i_1, \ldots, i_{n-1}, 1)$ denotes an n-path and there exists a k such that $i_k = i_{k+2}$ for some k then the path P is said to contain a 2-cycle. It was observed that many n-paths contained 2-cycles. They proposed a tighter relaxation by forbidding paths containing 2-cycles and presented a dynamic programming algorithm for finding an n-path of minimum cost which does not contain 2-cycles. Houck et al. [30] also showed that the problem of maximizing the bound derived by this relaxation could be expressed as linear programs similar to (68) and (69). They proposed a column generation scheme for solving the master problem but noted that the column generation was very slow in converging to the optimal solution. This is a typical observation in early research involving column generation. An American pioneer in linear programming computing techniques, Orchard-Hays [67, p. 240] said: “Nevertheless, the D–W (Dantzig–Wolfe) (generalized programming) algorithm presents difficulties, and overall experience with its use has been mixed. This has led to some disappointment with decomposition algorithms”. Houck et al. [30] then proposed to use the subgradient algorithm following the earlier approach of Held and Karp [24] and embedded this method in a branch and bound algorithm. The authors made an important concluding remark that the n-path relaxation can easily accommodate extra conditions. In fact the computational work required in the dynamic programming algorithm is just reduced when additional constraints are handled. In relation to this observation Picard and Queyranne [68] had previously considered a time-constrained variant of the TSP.

Christofides et al. [62,69] generalized the n-path relaxation of Houck et al. [30]. They formulated the capacitated VRP using resource-constrained paths where the resource is the accumulated demand q along the path and called these paths q-paths. They also considered the set-partitioning formulation of the CVRP similar to (69) but instead of solving the linear programming relaxation of this master problem they proposed a relaxation which could be solved by dynamic programming. In this way their method is a two-level dynamic programming approach. In 1987 Kolen et al. [31] introduced the first method for the exact solution of the VRPTW. Kolen et al. [31] extended the two-level dynamic programming approach of christofides et al. [62] for the VRPTW by introducing the accumulated time along the paths as an additional resource. The importance of this research comes from the introduction of the shortest path problem with time windows and capacity constraints SPPTWCC, which has played a prominent role in the research on the VRPTW. The method of Kolen et al. [31] was only capable of solving problems with up to 15 customers. The reason for this is the relaxation of the master level problem and the use of dynamic programming for solving the master problem.

The appearance of 'A new optimization algorithm for the VRP with time windows' [32] in the journal Operations Research was a breakthrough in the history of the VRPTW and furthermore an important paper in relation to the successful application of Dantzig–Wolfe decomposition and column generation in general. The method of Dosrochers et al. [32] is also based on the resource-constrained path formulation of the VRPTW but they used column generation to solve the linear programming relaxation of the set-partitioning master problem (69). The idea of embedding column generation in a branch and bound algorithm was previously introduced by Desrochers et al. [70] for the m-TSP with time windows. Another important contribution of Desrochers et al. [32] was the introduction of the set of test problems developed by Solomon [52]. The introduction of a standard set of test problems is important because it enables relative evaluation of competing algorithms. Of course different authors also need to consider the same problem variant and adhere to certain conventions with respect to the precision of problem data. Finally, the Desrochers et al. [32] paper introduced the label setting algorithm of Desrochers [66] in the context of the VRPTW for solving the shortest path problem with time windows and capacity constraints. The authors extended the algorithm to include the 2-cycle elimination scheme of Houck et al. [30]. Desrochers' algorithm has been an important component in the solution of a large class of resource constrained routing and scheduling problems [71] but as far as we are aware the manuscript 'An algorithm for the shortest path problem with resource constraints' [66] has not yet been published in an international journal.

Lagrangian decomposition is an approach that attempts to strengthen the bounds of Lagrangian relaxation [72]. This approach splits the original problem into two or more different types of subproblems. Halse [33] describes three different Lagrangian decompositions of the VRPTW: VS1, VS2, and VS3. In the VS1 decomposition the VRPTW is formulated using a shortest path problem with time windows (SPPTW) and a generalized assignment problem (GAP). The VS2 decomposition considers a shortest path problem with time windows and capacity constraints (SPPTWCC) and a semi-assignment problem (SAP). Finally, the VS3 decomposition considers an SPPTWCC and a GAP by including vehicle capacity constraints in both subproblems. The Lagrangian dual problems of the three Lagrangian decomposition algorithms involves more multipliers than in the Lagrangian relaxation of the VRPTW (61). In fact, if n is the number of vehicles then the Lagrangian decompositions requires n times as many multipliers as the Lagrangian relaxation (61). Kohl [73] made an analytical comparison of the bounds provided by the Lagrangian decompositions and the Lagrangian relaxation and proved that the VS1 and VS2 decompositions give the same bound as the Lagrangian relaxation. Furthermore, he also proved that the VS3 decomposition gives the same bound as the Lagrangian relaxation under the assumptions that the vehicles are identical and a feasible solution exists for the VRPTW. This proof is non-trivial since the two subproblems SPPTWCC and GAP do not have the integrality property. Since the Lagrangian decompositions increase the dimension of the Lagrangian dual problem, but do not strengthen the bound of the Lagrangian relaxation, the conclusion of the analysis by Kohl [73] is that the Lagrangian decompositions are not attractive for the identical-vehicle VRPTW. Fisher et al. [29] presented computational results based on the VS2 decomposition solving the Lagrangian dual problem using the subgradient algorithm. Although the Lagrangian decomposition considered by Fisher et al. [29] is not attractive compared to the Lagrangian relaxation the authors were the first to consider a path formulation of the VRPTW from the dual point of view. It would later become clear that it was the choice of the subgradient algorithm that impeded the dual approach.

Kohl and Madsen [34] proposed a method for the VRPTW based on the Lagrangian relaxation (61). They implemented a bundle algorithm of Lemaréchal et al. [74] with an Euclidean steepest descent direction finding problem and the line-search of Lemaréchal [75] for determining the step-size. In relation to line-searches Hiriart-Urruty and Lemaréchal [18, p. 403] mentioned that "the modern tendency goes towards the so-called trust-region technique". A number of important issues were addressed by Kohl and Madsen [34]. First, the dimension of the Lagrangian dual problem is smaller in the Lagrangian relaxation than in the Lagrangian decompositions. Second, the convergence of the bundle algorithm is better compared to the subgradient algorithm. Third, using a bundle algorithm it is possible to obtain a primal solution equivalent to the variables of the Dantzig–Wolfe master problem (69). Finally, in the bundle algorithm one can choose a starting point with relatively small multiplier values and gradually increase the multipliers to the optimal level. The dual approach therefore creates easier shortest path subproblems because the modified arc costs are less negative, hence less negative cycles are introduced. Kohl and Madsen [34] only considered problems that required very little branching so a full computational study of the path formulation from a dual point of view was not performed.

Kohl [73] and Kohl et al. [35] addressed the need for improving the bounds provided by the non-elementary path formulation with 2-cycle elimination. They introduced the k-path inequalities (13) for $k = 2$ in the Dantzig–Wolfe master problem (69). The path formulation then has exponentially many variables as well as constraints and a column and cut generation approach was used for solving the master problem. They embedded the column and cut generation in a branch and bound algorithm and their work is one of the early examples of what became known as branch, price, and cut algorithms.

Larsen [36] parallelized the branch and bound search in the algorithm of Kohl et al. [35]. Furthermore, Larsen [36,37] proposed column deletion and forced early stop for improving solution times. The column deletion procedure deletes columns in the master problem and is similar to the concept of the bundle reduction technique used by Kohl and Madsen [34], i.e. to limit the size of the coordinating master problem. The forced early stop terminates the SPPTWCC pricing algorithm as soon as one path with negative reduced cost is generated. The forced early stop is motivated from a dual point of view. In the column generation algorithm the initial master problem must be initialized with a feasible solution. If this primal solution corresponds to a dual solution with relatively high multiplier values compared to the optimal level then the difficulty of the SPPTWCC subproblem is relatively higher in the initial phase of the column generation algorithm. In the first iterations forced early stop therefore gives rapid improvements of the dual solution cutting down on solution times in the subproblem. Forced early stop is also called partial pricing whereas solving the subproblem to optimality is called full pricing.

Rich [76] and Cook and Rich [38] extended the work of Kohl et al. [35]. They proposed a new separation algorithm for the k-path inequalities for $k \geqslant 2$. The authors used the randomized algorithm given by Karger and Stein [77] that finds α-minimal cuts in undirected graphs in $O(n^{2\alpha}\log^2 n)$ time where n is the number of nodes. By setting $\alpha = k$, sets W for which $x(W) < k$ are found. If $k(W) \geqslant k$ in (7) then W induces a valid k-path inequality (13). Moreover, they parallelized the separation of k-path inequalities and the branch and bound search.

The bounds provided by the path formulation is also improved if we require the solution to the resource-constrained shortest path problem to be elementary. Beasley and Christofides [78] proposed to add a visitation resource for each node $v \in V_*$ with a lower and upper limit of zero and one, respectively. The visitation resource usage when passing through node v is one. This new resource definition ensures we generate only elementary paths. However, the size of the state space increases dramatically and Beasley and Christofides [78] expected that this formulation would only be suitable for solving problems with a small number of resources and made no further studies. The approach of Desrochers et al. [32] is attractive from a computational point of view because the SPPTWCC can be solved in pseudo-polynomial time. However, the method has the disadvantage of weakening the lower bound provided by the path formulation. In an effort to address this issue Feillet et al. [39], Chabrier [40], and Danna and Le Pape [41] adapted the idea of Beasley and Christofides [78] to Desrochers' algorithm. The authors also generalized the consumption of the visitation resource. A node v is called unreachable with respect to a path P if the path already has visited v or there is no way to extend the path to v due to other resource limitations, i.e. time or capacity. The visitation resources of a path are therefore consumed either because the nodes have already been visited or because of other resource constraints. The concept of unreachable nodes is attractive because it sharpens the dominance relation. Chabrier [40], Danna and Le Pape [41], and Feillet et al [42] proposed other heuristic and exact improvements incorporated in their algorithms for the solution of the elementary shortest path problem. The developments in the solution of the subproblem are interesting because the VRPTW polytope in the elementary approach is embedded in a larger polytope but not a polytope over which $c^T x$ can be minimized in polynomial time. In fact, the authors propose to solve another NP-complete problem instead of the VRPTW. In our view this raises the following research question: Can one find an algorithm for the VRPTW that compute polynomial-time lower bounds that are not dominated by the bounds obtained by the elementary path formulation? We believe that further investigations of the VRPTW polytope $\mathscr{P}_{\text{VRPTW}}$ of Definition 2.13 will prove valuable in answering this question.

Kallehauge [79] presented some measurements of the effects of relaxing the condition that paths must be elementary. The measurements followed a suggestion by Natashia Boland, who raised the following research question: What are the gaps between the solutions to (69) with elementary routes and solutions to (69) with non-elementary routes and 2-cycle elimination for Solomon's data sets? She suggested that a good start would be to look at the non-elementary LP solutions that are produced, and check how many non-elementary paths are assigned non-zero LP values, and measuring the sum of the non-elementary LP variables over the sum of the LP-variables, i.e. the fraction of the non-elementary

Table 1
Measuring the effects of the non-elementary relaxation in the root node before any inequalities are generated

	NE CUSTOMERS	NE PATHS/ ALL PATHS	NE FLOW/ TOTAL FLOW	%NE FLOW
R101	0	0/26	0/19.5	0
R102	0	0/18	0/18.0	0
R103	8	7/39	1.22/14.06	8.7
R104[a]	49	41/78	4.18/10.16	41.1
R105	0	0/59	0/14.88	0
R106	8	9/74	1.37/13.00	10.5
R107	23	29/78	2.42/10.95	22.1
R108[a]	34	32/73	3.63/9.82	37.0
R109	14	14/78	1.70/12.23	14.0
R110	24	25/89	2.86/10.96	26.1
R111	17	20/91	1.87/11.43	16.4
R112[a]	41	36/86	2.79/9.49	29.4
RC101	0	0/60	0/14.58	0
RC102	11	10/61	1.88/13.51	14.0
RC103	20	24/72	3.11/10.69	29.0
RC104[a]	53	48/85	4.22/9.88	42.7
RC105	2	2/35	0.40/13.53	3.0
RC106[a]	29	32/89	3.42/12.34	27.7
RC107[a]	31	30/74	2.87/11.48	25.0
RC108[a]	40	38/65	3.58/10.73	33.4

[a]Unsolved instance by September 2000.

paths. Kallehauge [80] made these measurements for all Solomon's short-horizon 100 customer problems (excluding C1 problems) and it showed that the problems that remained unsolved have a high fraction of flow on non-elementary paths. In Table 1 we present these measurements, which are made in the root node of the branch-and-bound tree before inserting any cuts. NE CUSTOMERS is the number of customers that are visited more than once. NE PATHS/ ALL PATHS is the number of corresponding paths in the LP-solution that contain cycles compared to the total number of paths, i.e. the number of variables greater than zero. NE FLOW/ TOTAL is the sum of the LP-variables that correspond to paths with cycles and the total sum. %NE FLOW shows the percentage of the non-elementary compared to the total flow.

Following a suggestion by Jacques Desrosiers, Kallehauge [80] presented the same measurements as in Table 1 after the total flow is integer, i.e. when the number of vehicles is integer. It is also possible to include e.g. subtour inequalities and 2-path inequalities in the LP-model before branching on vehicles. Table 2 shows these measurements after subtour inequalities and 2-path inequalities are generated for the LP problem of the root node and we have branched on vehicles. We keep generating these inequalities as long as we only branch on vehicles. The number of times we branch on vehicles is relatively small because the number of vehicles quickly becomes integer. If we compare the column %NE FLOW in Table 1 with the equivalent column in Table 2 we see that the flow on the non-elementary paths is decreased after inserting cuts and branching on vehicles but it is still significant, i.e. above 20% for all unsolved instances.

Table 3 shows for each instance the number of k-cycles on the paths with non-zero LP-values in the optimal solution to the LP problem of the root node. k-CYCLES shows the number of cycles by type: 3-cycle/4- cycle/5-cycle/etc. It is characteristic for the instances that remained unsolved that a high fraction of the total number of cycles in the non-elementary LP solution is 3-cycles. Irnich [81] generalized the 2-cycle elimination of Houck et al. [30] and Irnich and Villeneuve [43] further extended this work with detailed computational results available in [82]. Indeed the k-cycle elimination gave improvements in the lower bounds and was still computationally attractive as long as k is not too large. Irnich and Villeneuve [43] presented computational results for values of $k=2,\ldots,5$. Irnich and Villeneuve [43] gave an upper bound on the increase in the number of labels. This bound is $(k-1)!^2$ for $k>3$ so there are limits to how large values of k should be used. For $k=2$ the increased number of labels is a factor 2, for $k=3$ it is a factor 6.

Kallehauge et al. [44] presented a stabilized cutting-plane algorithm for the Lagrangian dual problem. The idea is to force the next dual solution of the cutting-plane algorithm to be a priori in a ball or trust-region associated with the given norming. The authors use the max-norm so the master problem is an LP problem with bounds on the dual variables. This is an acceleration of the cutting-plane algorithm of Kelley [83] and Cheney and Goldstein [84]; the

Table 2
Measuring the effects of the non-elementary relaxation after inequalities are generated and branching on vehicles

	NE CUSTOMERS	NE PATHS/ ALL PATHS	NE FLOW/ TOTAL FLOW	%NE FLOW
R101	0	0/29	0/20.00	0
R102	0	0/18	0/18.00	0
R103	4	4/42	1.00/15.00	6.7
R104[a]	45	29/81	2.95/11.00	26.8
R105	0	0/74	0/15.00	0
R106	6	7/74	1.11/13.00	8.5
R107	16	20/87	1.69/12.00	14.1
R108[a]	35	35/85	2.90/10.00	29.0
R109	6	6/65	0.85/13.00	6.5
R110	23	23/92	2.51/11.00	22.8
R111	18	19/90	1.97/12.00	16.4
R112[a]	40	31/91	2.70/10.00	27.0
RC101	0	0/77	0/16.00	0
RC102	12	19/97	1.91/14.00	13.6
RC103	22	25/85	2.97/11.00	27.0
RC104[a]	42	37/93	2.93/10.00	29.3
RC105	0	0/25	0/15.00	0
RC106[a]	28	30/99	3.03/13.00	23.3
RC107[a]	37	35 /101	2.85/12.00	23.8
RC108[a]	40	48/97	2.95/11.00	26.8

[a]unsolved instance by September 2000.

Table 3
Number of k-cycles (3-cycles/4-cycles/5-cycles/etc.) on paths with non-zero LP-values in root node

	k-CYCLES
R101	0
R102	0
R103	4/1/1/1/1/0/0/1
R104[a]	29/7/11/10/6/6/8/4/7
R105	0
R106	1/0/1/1/3/3
R107	7/5/1/9/6/5/3/2
R108[a]	26/2/4/5/2/8/6/8/5
R109	19/0/1
R110	24/4/2/1/2/0/1
R111	10/4/3/0/2/1/2/1
R112[a]	54/7/14/8/0/0/0/1
RC101	0
RC102	5/2/2/1/2/2/3/1
RC103	18/5/6/3/2/1/2
RC104[a]	55/9/8/9/5/9/6/7/4
RC105	2
RC106[a]	36/8
RC107[a]	38/3/6/3/3/2
RC108[a]	55/7/3/1/5/1/3/3

[a]Unsolved instance by September 2000.

original reference for the column generation variant is Dantzig and Wolfe [85]. The trust-region ensures stability of the dual solution from one iteration to the next. Instability refers to the situation where the current iterate is closer (with respect to some norm) to the optimal solution than the next iterate. Kallehauge et al. [44] was motivated by the work on acceleration strategies at the master problem level by Kohl and Madsen [34]. However, using the simple max-norm trust-region method the authors avoided solving the quadratic problems of the version of the bundle algorithm Kohl and Madsen [34] considered and they also avoided the line-searches associated with the bundle algorithm. To obtain feasible integer solutions the cutting-plane algorithm is embedded in the branch, cut, and price framework ABACUS [20]. ABACUS is a C++ class library for solving mixed-integer linear-programs (MILP) by branch, cut, and price. It is interesting to note that the authors' formulation of the MILP presented to ABACUS only involves continuous variables. Obviously, that is because it is the dual problem (68) that is presented to ABACUS. Branching decisions are then based on the dual variables of the master problem, i.e. the path variables of (69). The authors also introduce inequalities in the master problem. Because the master problem is stated on the dual variables, subtour and 2-path inequalities are added as columns to this problem. Thienel [19] noted that although ABACUS was designed for linear programming relaxations there were no reasons that the branch and bound algorithm of ABACUS was restricted to this type of relaxation. However, Thienel [19] thought that it would require a generalization of ABACUS to use the system for Lagrangian relaxation. In fact Kallehauge et al. [44] showed that by remaining within the context of linear programming when solving the Lagrangian dual problem it is already possible to embed Lagrangian relaxation in this system. The trust-region method of Kallehauge et al. [44] can be used to solve anyLagrangian dual problem associated with a hard optimization problem and its implementation in ABACUS would allow the developer to concentrate on the problem specific parts, i.e. the cutting plane and the column generation, the branching rules, and the primal heuristics.

Recently Jepsen et al. [46] introduced 3-customer clique inequalities, called subset row (SR) inequalities, that are valid for the set-partitioning polytope of (69). These inequalities change the structure of the shortest path subproblem and the authors describe how the dominance relation of the subproblem is modified to incorporate these clique inequalities. This is the first example of strengthening the path formulation by introducing inequalities defined directly with respect to the path variables of the master problem. The authors solve the elementary shortest path subproblem with a bi-directional label setting algorithm developed by Righini and Salani [45] who showed that the bi-directional dynamic programming algorithm outperforms the mono-directional algorithm previously used in the literature for resource constrained shortest path computation. Jepsen et al. [46] also showed that the SR-inequalities can be used in the context of the k-cycle algorithm of Irnich and Villeneuve [43]. The authors note that this is particularly interesting for larger VRPTW instances, since preliminary results show that the elementary shortest path algorithm is considerably slower than the shortest path algorithm with k-cycle elimination when the number of customers increases.

7. Conclusions

In this paper we have reviewed four different formulations of the VRPTW and the exact algorithms associated with them. We have identified and organized a total of 21 references on the VRPTW relative to four seminal papers on formulations of the TSP: arc formulation, arc-node formulation, spanning tree formulation, and path formulation. Out of these 21 references 17 references are related to the path formulation of the VRPTW. The polyhedral approach of the arc formulation is in our opinion promising and relatively little research has been conducted along these lines compared to the decomposition approach of the path formulation. Furthermore, the spanning tree formulation of the VRPTW has not been the subject of the same amount of research as the path formulation and the extension of the shortest spanning tree subproblem to time windows has not been considered in the literature. In our view it is at this point therefore an open question whether the spanning tree formulation described in this paper is effective compared to the path formulation.

The exact algorithms based on the path formulation has been very successful and the most important contributions of the research on the VRPTW lies in this area. The developments in the solution of the subproblem are interesting because the VRPTW polytope in the elementary path formulation approach is embedded in a larger polytope but not a polytope over which the objective function can be minimized in polynomial time. In fact, the authors propose to solve another NP-complete problem instead of the VRPTW, i.e. a resource-constrained elementary shortest path problem. In our view this raises the question of one can find an algorithm for the VRPTW that compute polynomial-time lower bounds that are not dominated by the bounds obtained by the elementary path formulation. We believe that further investigations of the VRPTW polytope will prove valuable in answering this question. Furthermore, the recent development of a bi-directional dynamic programming algorithm for resource constrained shortest path problems is also a significant contribution. The developments in the solution of the (dual) master problem associated with the path formulation are also very interesting. There are at least three important developments. First, introduction of strong valid inequalities for the VRPTW polytope in the master problem, e.g. generalized subtour inequalities. Second, development of acceleration techniques that addresses the instability issues with the cutting-plane algorithm for convex minimization or equivalently the column generation algorithm of the Dantzig–Wolfe decomposition. Third, and more recently, strong valid inequalities have been introduced for the set-partitioning polytope and thereby strengthening the lower bounds provided by this relaxation. The inequalities can also be used from a dual point of view in the Lagrangian dual problem.

It is clear that 'A new optimization algorithm for the vehicle routing problem with time windows' [32] was a very substantial achievement. It is important both for introducing the path formulation and the column generation algorithm to the VRPTW and for the future developments it inspired. It is remarkable how much of the scope and methodology of combinatorial optimization has been applied in the attack on the VRPTW. The importance of the research described in this paper comes not from the number of applications where the mathematical model of the VRPTW precisely fits, but from the fact that the vehicle routing problem with time windows is typical of other resource-constrained problems in combinatorial optimization.

References

[1] Dantzig GB, Ramser JH. The truck dispatching problem. Management Science 1959;6:80–91.
[2] Toth P, Vigo D, editors. The vehicle routing problem, SIAM monographs on discrete mathematics and applications. Philadelphia: SIAM; 2002.
[3] Toth P, Vigo D. Branch-and-bound algorithms for the capacitated VRP. In: Toth P, Vigo D, editors. The vehicle routing problem, SIAM monographs on discrete mathematics and applications. Philadelphia: SIAM; 2002. p. 29–51.
[4] Naddef D, Rinaldi G. Branch-and-cut algorithms for the capacitated VRP. In: Toth P, Vigo D, editors. The vehicle routing problem SIAM monographs on discrete mathematics and applications. Philadelphia: SIAM; 2002. p. 49–78.
[5] Bramel J, Simchi-Levi D. Set-covering-based algorithms for the capacitated VRP. In: Toth P, Vigo D, editors. The vehicle routing problem SIAM monographs on discrete mathematics and applications. Philadelphia: SIAM; 2002. p. 85–108.
[6] Cordeau JF, Desaulniers G, Desrosiers J, Solomon MM, Soumis F. VRP with time windows. In: Toth P, Vigo D, editors. The vehicle routing problem SIAM monographs on discrete mathematics and applications. Philadelphia: SIAM; 2002. p. 157–93.
[7] Toth P, Vigo D. VRP with backhauls. In: Toth P, Vigo D, editors. The vehicle routing problem SIAM monographs on discrete mathematics and applications. Philadelphia: SIAM; 2002. p. 195–224.
[8] Desaulniers G, Desrosiers J, Erdmann A, Solomon MM, Soumis F. VRP with pickup and delivery. In: Toth P, Vigo D, editors. The vehicle routing problem SIAM monographs on discrete mathematics and applications. Philadelphia: SIAM; 2002. p. 225–42.
[9] Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB, editors. The traveling salesman problem—a guided tour of combinatorial optimization. New York: Wiley; 1985.
[10] Hoffman AJ, Wolfe P. History. In: Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB, editors. The traveling salesman problem—a guided tour of combinatorial optimization. New York: Wiley; 1985. p. 1–15.
[11] Dantzig G, Fulkerson R, Johnson S. Solution of a large-scale traveling-salesman problems. Operations Research 1954;2:393–410.
[12] Padberg M, Rinaldi G. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Review 1991;33:60–100.
[13] Laporte G, Nobert Y, Desrochers M. Optimal routing under capacity and distance restrictions. Operations Research 1985;33:1050–73.
[14] Schrijver A. Combinatorial optimization, polyhedra and efficiency, algorithms and combinatorics 24. Berlin Heidelberg: Springer; 2003.
[15] Nemhauser GL, Wolsey LA. Integer and combinatorial optimization. New York: Wiley; 1988.
[16] Shapiro JF. Mathematical programming: structures and algorithms. New York: Wiley; 1979.
[17] Everett III. H. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. Operations Research 1963;11:399–417.
[18] Hiriart-Urruty JB, Lemaréchal C. Convex analysis and minimization algorithms I-II, grundlehren der matematischen wissenschaften 304-305. Berlin Heidelberg: Springer; 1993.
[19] Thienel S, ABACUS a branch-and-cut system. PhD thesis, Mathematisch-Naturwissenschaftlichen Fakultät, Universität zu Köln; 1995.
[20] Jünger M, Thienel S. The ABACUS system for branch-and-cut-and-price algorithms in integer programming and combinatorial optimization. Software: Practice and Experience 2000;30:1325–52.
[21] Ralphs T, Guzelsoy M. The SYMPHONY callable library for mixed integer programming. Technical Report, Department of Industrial and Systems Engineering, Leheigh University; 2004.
[22] COIN/BCP User's manual. 2001.
[23] Held M, Karp RM. The traveling-salesman problem and minimum spanning trees. Operations Research 1970;18:1138–62.
[24] Held M, Karp RM. The traveling-salesman problem and minimum spanning trees: Part II. Mathematical Programming 1971;1:6–25.
[25] Kallehauge B, Boland N, Madsen OBG. Path inequalities for the vehicle routing problem with time windows. Networks (2006), to appear.
[26] Mak V, Ernst A. New cutting planes for the time and/or precedence constrained asymmetric travelling salesman problem and vehicle routing problem. Mathematical Methods of Operations Research (2006), to appear.
[27] Miller CE, Tucker AW, Zemlin RA. Integer programming formulation of traveling salesman problems. Journal of the Association for Computing Machinery 1960;7:326–9.
[28] Bard JF, Kontoravdis G, Yu G. A branch-and-cut procedure for the vehicle routing problem with time windows. Transportation Science 2002;36:250–69.
[29] Fisher ML, Jörnsten KO, Madsen OBG. Vehicle routing with time windows: two optimization algorithms. Operations Research 1997;45: 488–92.

[30] Houck Jr. DJ, Picard JC, Queyranne M, Vemuganti RR. The travelling salesman problem as a constrained shortest path problem: theory and computational experience. OPSEARCH 1980;17:93–109.

[31] Kolen AWJ, Rinnooy Kan AHG, Trienekens HWJM. Vehicle routing with time windows. Operations Research 1987;35:266–73.

[32] Desrochers M, Desrosiers J, Solomon M. A new optimization algorithm for the vehicle routing problem with time windows. Operations Research 1992;40:342–54.

[33] Halse K, Modeling and solving complex vehicle routing problems. PhD thesis, Department of Mathematical Statistics and Operations Research, Technical University of Denmark; 1992.

[34] Kohl N, Madsen OBG. An optimization algorithm for the vehicle routing problem with time windows based on Lagrangean relaxation. Operations Research 1997;45:395–406.

[35] Kohl N, Desrosiers J, Madsen OBG, Solomon MM, Soumis F. 2-path cuts for the vehicle routing problem with time windows. Transportation Science 1999;33:101–16.

[36] Larsen J, Parallelization of the vehicle routing problem with time windows. PhD thesis, Department of Mathematical Modelling, Technical University of Denmark; 1999.

[37] Larsen J. Refinements of the column generation process for the vehicle routing problem with time windows. Journal of Systems Science and Systems Engineering 2004;13:326–41.

[38] Cook W, Rich JL. A parallel cutting-plane algorithm for the vehicle routing problem with time windows. Technical Report, Rice University; 1999.

[39] Feillet D, Dejax P, Gendreau M, Gueguen C. An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. Networks 2004;44:216–29.

[40] Chabrier A. Vehicle routing problem with elementary shortest path based column generation. Computers & Operations Research 2006;33: 2972–90.

[41] Danna E, Le Pape C. Branch-and-price heuristics: a case study on the vehicle routing problem with time windows. In: Desaulniers G, Desrosiers J, Solomon MM, editors. Column generation, GERAD 25th Anniversary Series. New York: Springer; 2005. p. 99–129.

[42] Feillet D, Gendreau M, Rousseau LM. New refinements for the solution of vehicle routing problems with branch and price. Technical Report C7PQMR PO2005-08-X, Center for Reasearch on Transportation, Montreal; 2005.

[43] Irnich S, Villeneuve D. The shortest-path problem with resource constraints and k-cycle elimination for $k \geqslant 3$. INFORMS Journal on Computing 2006;18:16.

[44] Kallehauge B, Larsen J, Madsen OBG. Lagrangian duality applied to the vehicle routing problem with time windows. Computers & Operations Research 2006;33:1464–87.

[45] Righini G, Salani M. Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optimization 2006;3:255–73.

[46] Jepsen M, Spoorendonk S, Petersen B, Pisinger D. A non-robust branch-and-cut-and-price algorithm for the vehicle routing problem with time windows. Technical Report 06/03, Department of Computer Science, University of Copenhagen; 2006.

[47] Kallehauge B, Larsen J, Madsen OBG, Solomon MM. Vehicle routing problem with time windows. In: Desaulniers G, Desrosiers J, Solomon MM, editors. Column generation, GERAD 25th Anniversary Series. New York: Springer; 2005. p. 67–98.

[48] Garey MR, Johnson DS. Computers and intractability, a guide to the theory of NP-completeness. New York: Freeman; 1979.

[49] Ascheuer N, Fischetti M, Grötschel M. A polyhedral study of the asymmetric traveling salesman problem with time windows. Networks 2000;36:69–79.

[50] Mak VH. On the asymmetric travelling salesman problem with replenishment arcs. PhD thesis, Department of Mathematics and Statistics, The University of Melbourne; 2001.

[51] Balas E, Fischetti M, Pulleyblank WR. The precedence-constrained asymmetric traveling salesman polytope. Mathematical Programming 1995;68:241–65.

[52] Solomon MM. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 1987;35: 254–65.

[53] Grötschel M, Padberg MW. Polyhedral theory. In: Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB, editors. The traveling salesman problem—a guided tour of combinatorial optimization. New York: Wiley; 1985. p. 251–305.

[54] Kulkarni RV, Bhave PR. Integer programming formulations of vehicle routing problems. European Journal of Operational Research 1985;20: 58–67.

[55] Desrochers M, Laporte G. Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. Operations Research Letters 1991;10:27–36.

[56] Ascheuer N, Fischetti M, Grötschel M. Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. Mathematical Programming Series A 2001;90:475–506.

[57] Papadimitriou CH. The complexity of the capacitated tree problem. Networks 1978;8:217–30.

[58] Gilmore PC, Gomory RE. A linear programming approach to the cutting-stock problem – part II. Operations Research 1963;11:863–88.

[59] Held M, Wolfe P, Crowder P. Validation of subgradient optimization. Mathematical Programming 1974;6:62–88.

[60] Beasley JE. Lagrangean relaxation. In: Reeves CR, editor. Modern heuristic techniques for combinatorial problems, advanced topics in computer science. London: McGraw-Hill; 1995. p. 243–303.

[61] Geoffrion AM. Lagrangian relaxation for integer programming. Mathematical Programming Study 1974;2:82–114.

[62] Christofides N, Mingozzi A, Toth P. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. Mathematical Programming 1981;20:255–82.

[63] Fisher ML. Optimal solution of vehicle routing problems using minimum K-trees. Operations Research 1994;42:626–42.

[64] Toth P, Vigo D. An exact algorithm for the capacitated shortest spanning arborescence. Annals of Operations Research 1995;61:121–41.

[65] Dror M. Note on the complexity of the shortest path models for column generation in VRPTW. Operations Research 1994;42:977–8.
[66] Desrochers M. An algorithm for the shortest path problem with resource constraints. Technical Report, GERAD; 1988.
[67] Orchard-Hays W. Advanced linear-programming computing techniques. New York: McGraw-Hill; 1968.
[68] Picard JC, Queyranne M. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. Operations Research 1978;26:86–110.
[69] Christofides N, Mingozzi A, Toth P. State-space relaxation procedures for the computation of bounds to routing problems. Networks 1981;11: 145–64.
[70] Desrosiers J, Soumis F, Desrochers M. Routing with time windows by column generation. Networks 1984;14:545–65.
[71] Desrosiers J, Dumas Y, Solomon MM, Soumis F. Time constrained routing and scheduling, In: Ball M, Magnanti T, Monma C, Nemhauser GL, editors. Network routing. Amsterdam, The Netherlands: Elsevier Science Publishers B.V. (North-Holland); 1995, vol. 8 Handbooks in operations research and management science. p. 35–139.
[72] Guignard M, Kim S. Lagrangean decomposition: a model yielding stronger Lagrangean bounds. Mathematical Programming 1987;39:215–28.
[73] Kohl N, Exact methods for time constrained routing and related scheduling problems, PhD thesis, Department of Mathematical Modelling, Technical University of Denmark; 1995.
[74] Lemaréchal C, Strodiot JJ, Bihain A. On a bundle algorithm for non-smooth optimization. In: Mangasarian OL, Meyer RR, Robinson SM, editors. Nonlinear programming 4. New York: Academic Press; 1981. p. 245–82.
[75] Lemaréchal C, A view of line-searches. In: Auslender A, Oettli W, Stoer J, editors. Optimization and optimal control. Berlin Heidelberg: Springer; 1981, vol. 30. Lecture notes in control and information sciences. p. 59–78.
[76] Rich JL. A computational study of vehicle routing applications. PhD thesis, Rice University; 1999.
[77] Karger DR, Stein C. A new approach to the minimum cut problem. Journal of the Association for Computing Machinery 1996;43:601–40.
[78] Beasley JE, Christofides N. An algorithm for the resource constrained shortest path problem. Networks 1989;19:379–94.
[79] Kallehauge B. A hybrid optimal method to the vehicle routing problem with time windows. In: ROUTE 2000 - International workshop on vehicle routing. Skodsborg, Denmark, August 16–19, 2000. Presentation available at ⟨http://www1.ctt.dtu.dk/ROUTE2003/route2000/presentations/presentation_brian.pdf⟩.
[80] Kallehauge B. Ideas for solving VRPTW. September 21, 2000. Private communication to Jacques Desrosiers, Madsen OBG and Jesper Larsen.
[81] Irnich S. The shortest path problem with k-cycle elimination ($k \geqslant 3$): Improving a branch and price algorithm for the VRPTW. Technical Report, Lehr- und Forschungsgebiet Unternehmensforschung, Rheinisch-Westfälische Technische Hochschule, November 22, 2000.
[82] Irnich S, Villeneuve D. Online supplement for the shortest-path problem with resource constraints and k-cycle elimination for $k \geqslant 3$. ⟨http://joc.pubs.informs.org/OnlineSupplements.html⟩.
[83] Kelley JE. The cutting-plane method for solving convex programs. Journal of SIAM 1960;8:703–12.
[84] Cheney EW, Goldstein AA. Newton's method for convex programming and tchebycheff approximation. Numerische Mathematik 1959;1: 253–68.
[85] Dantzig GB, Wolfe P. A decomposition principle for linear programs. Operations Research 1960;8:101–11.