ELSEVIER European Journal of Operational Research 105 (1998) 390–400

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodology

The rural postman problem with deadline classes

A.N. Letchford, R.W. Eglese *

Department of Management Science, The Management School, Lancaster University, Lancaster LA1 4YX, UK

Received 1 August 1995; accepted 1 February 1997

Abstract

Vehicle routing problems with general time windows are extremely difficult to solve. However, the time windows in a particular problem may have a special structure which can be exploited. We consider a single-vehicle arc-routing problem in which the arcs are partitioned into *deadline classes*. It is shown that a cutting-plane approach works well for this problem. © 1998 Elsevier Science B.V.

Keywords: Rural postman problem; Time windows; Integer programming; Cutting-planes

1. Introduction

In real-life vehicle routing problems, it frequently happens that the customers require service (e.g., deliveries) to occur within specified time windows (see Desrosiers et al., 1995). In the node-routing case (when the customers are located at the vertices of a network), the standard problem is known as the *Vehicle Routing Problem with Time Windows* or VRPTW. Large VRPTW instances cannot as yet be solved to optimality unless the windows are rather narrow (Kohl and Madsen, 1995). Even the single-vehicle version, the *Travelling Salesman Problem with Time Windows*, presents a challenge (Dumas et al., 1995).

The time window model is, however, unnecessarily general for a significant proportion of real-life routing problems. For example, it occasionally happens that every time window begins at time zero; i.e., the customers only give deadlines (Nygard et al., 1988; Thangiah et al., 1994). In such cases the time windows tend to be large and the approaches of Kohl and Madsen (1995) and others would result in unacceptable running times. Yet, no optimisation algorithm is currently available which is specifically tailored to the deadline case.

In some problems, the time windows are even more structured. For example, the customers may be partitioned into a small number of classes according to priority, with each class having its own deadline. This is typical for problems of parcel delivery (Stephenson, 1996), where, for example, deliveries might have to be made within an hour, a half-day or a day, depending upon the rate paid by the sender. Another example is found in Li and Eglese (1996) and the references therein, where a fleet of vehicles must treat roads with salt to prevent them from freezing. Motorways and some A-roads must be treated within 2 hours of a call-out, other A-roads

* Corresponding author. Fax: +44-1524 844 885; E-mail: r.eglese@lancaster.ac.uk.

PII S0377-2217(97)00090-8

within 4 hours, etc. This latter problem is an *Arc Routing Problem* or ARP (see Eiselt et al., 1995a,b), where it is the edges of the network which require service rather than the vertices.

This paper is concerned with a single-vehicle ARP involving such structured deadlines, which we call the *Rural Postman Problem with Deadline Classes* or RPPDC. The RPPDC contains the NP-Hard *Rural Postman Problem* (Orloff, 1974; Lenstra and Rinnooy Kan, 1976), as a special case. An optimisation algorithm is presented for the RPPDC, based on the use of valid inequalities as cutting-planes. This approach has worked well on other routing problems such as the well-known TSP (see, e.g. Applegate et al., 1994), the RPP (Corberán and Sanchis, 1994) and the GRP (Corberán and Sanchis, 1996; Letchford, 1996, 1997).

The remainder of the paper is organised as follows. Formal definitions are given in Section 2, along with an integer programming formulation. In Section 3, several classes of valid inequalities are given for this formulation. In Section 4, some separation algorithms are given, i.e., algorithms for detecting when valid inequalities are violated by a given LP relaxation. Some computational results are given in Section 5. Finally, concluding comments are given in Section 6.

2. Definitions and formulation

Suppose we are given a loopless, connected, undirected graph G with vertex set $V = \{1,\dots,N\}$ and edge set E, with $\{1\}$ representing the depot. There is a set $R \subseteq E$ of edges requiring service, which in turn is partitioned into a small number of *deadline classes* $R^1, \dots, R^L$ according to priority. Edges in R^1 must be serviced by time T^1, edges in R^2 must be serviced by T^2, and so on. The vehicle takes time t_e to traverse any $e \in E$ without servicing it, and time s_e to traverse any $e \in R$ while servicing it. The RPPDC is the problem of finding a feasible route which minimises cost, given that each time an edge $e \in E$ is traversed, a cost $c_e > 0$ is incurred. To be feasible, the route must start and end at the depot and each edge must be serviced within its time limit. Splitting of demands (partially servicing an edge and revisiting it later to complete the servicing) is not permitted.

When the deadlines are removed, the standard RPP is obtained. Although the RPP is NP-Hard, the RPPDC is in some sense even harder in that it is NP-Complete to decide whether or not a feasible solution exists to any given RPPDC instance, even when $L = 1$. This can easily be shown by reduction from the well-known *Hamiltonian Path Problem* (see Garey and Johnson, 1979). The algorithm described in this paper either yields an optimal solution or proves that the problem is infeasible.

The RPPDC can be formulated as an integer programme in a number of ways. For example, it is not hard to adapt the single commodity-flow formulation of the VRPTW (e.g., Desrosiers et al., 1995), to the RPPDC. Observe, however, that G will often be sparse in practice. Indeed, if the edges represent roads, then G will be planar or near-planar. So in this paper flow variables have been avoided in favour of an approach which can exploit sparsity in G.

Let R_A^B, with $1 \le A \le B \le L$, denote $R^A \cup \cdots \cup R^B$. Regard a feasible RPPDC route as being composed of L *time-phases*: in phase 1, the vehicle leaves the depot and services the edges in R^1 plus (optionally) some edges in R_2^L. There is then a *phase transition* to phase 2, which must occur no later than time T^1. The vehicle begins phase 2 at the vertex at which it ended phase 1, and services any remaining edges in R^2 plus (optionally) some remaining edges in R_3^L. And so on. When the final phase, L, is finished (no later than T^L), all edges have been serviced and the vehicle returns to the depot via a shortest path.

The above observations lead to the following variable definitions:

x_{ep} = number times $e \in E$ is traversed *without* servicing in time phase p.
y_{ep} = 1 if $e \in R$ is serviced in time phase p, 0 otherwise.
z_{vp} = 1 if time phase p ends at $v \in V$, 0 otherwise.

The x_{ep} are defined for all $e \in E$, $p = 1,\dots,L$. For a given $e \in R^m$, y_{ep} is defined for $p = 1,\dots,m$. When $e \in R^1$, however, y_{ep} need not be defined since it must equal 1. The z_{vp} are defined for all $v \in V$, $p = 1,\dots,L$ and represent the phase transitions (when $p = 1,\dots,L-1$), or the return trip to the depot from the last edge serviced (when $p = L$). Certain z_{vp} can in fact be fixed to zero without losing generality (e.g., by assuming that the vehicle finishes phase 1 immediately after servicing an edge in R^1), but we define all z_{vp} for notational convenience.

We will also need the following notation: let c_v^* denote the cost of the shortest path from vertex v to the depot. For any $S \subset V$, let $\delta(S)$ (respectively, $E(S)$) denote the set of edges having exactly one end-vertex (exactly two end-vertices) in S. For any $F \subseteq E$, let $R_A^B(F) = R_A^B \cap F$ and $x_A^B(F) = \sum_{p=A}^{B} \sum_{e \in F} x_{ep}$. For any $F \subseteq R_A^L$, let $y_A^B(F) = \sum_{p=A}^{B} \sum_{e \in F} y_{ep}$. If $A = B$ in any of the above expressions, we drop the subscript, writing $R^B(E(S))$, $R^B(\delta(S))$ and so on. Finally, for any $\kappa \subseteq V$, let $z^p(\kappa) = \sum_{v \in \kappa} z_{vp}$.

Now note that it is never optimal for the vehicle to traverse an edge more than twice in any particular phase, since an Eulerian multigraph with three or more parallel edges remains Eulerian when two of these edges are removed. The RPPDC can now be formulated as follows:

$$\text{Minimise} \sum_{p=1}^{L} \sum_{e \in E} c_e x_{ep} + \sum_{v \in V} c_v^* z_{vL}$$

subject to:

$$y_1^m(e) = 1 \quad (m = 2,\dots,L,\ e \in R^m), \tag{1}$$

$$\sum_{p=1}^{m} \left[\sum_{e \in E} t_e x_{ep} + \sum_{e \in R_{m+1}^L} s_e y_{ep} \right] \le T^m - \sum_{e \in R_1^m} s_e \quad (m = 1,\dots,L), \tag{2}$$

$$z^p(V) = 1 \quad (p = 1,\dots,L), \tag{3}$$

$$x^1(\delta(S)) + y^1(R_2^L(\delta(S))) + z^1(S) \ge 2 \quad (\forall S \subseteq V \setminus \{1\} : R^1(E(S)) \ne \varnothing,\ R^1(\delta(S)) = \varnothing), \tag{4a}$$

$$x^1(\delta(S)) + y^1(R_2^L(\delta(S))) + z^1(S) \ge 2 y_{e1} \quad (\forall S \subseteq V \setminus \{1\} : R^1(\delta(S) \cup E(S)) = \varnothing,\ e \in R_2^L(E(S))), \tag{4b}$$

$$x^1(\delta(S)) + y^1(R_2^L(\delta(S))) + z^1(S) \ge x_{e1} \quad (\forall S \subseteq V \setminus \{1\} : R^1(\delta(S) \cup E(S)) = \varnothing,\ e \in E(S)), \tag{4c}$$

$$x^p(\delta(S)) + y^p(R_p^L(\delta(S))) + z^{p-1}(S) + z^p(S) \ge 2 y_{ep} \quad (\forall S \subseteq V,\ p = 2,\dots,L,\ e \in R_p^L(E(S))), \tag{4d}$$

$$x^p(\delta(S)) + y^p(R_p^L(\delta(S))) + z^{p-1}(S) + z^p(S) \ge x_{ep} \quad (\forall S \subseteq V,\ p = 2,\dots,L,\ e \in R_p^L(E(S))), \tag{4e}$$

$$x^1(\delta(v)) + y^1(R_2^L(\delta(v))) + z_{v1} \equiv |R^1(\delta(v))| \bmod 2 \quad (\forall v \in V \setminus \{1\}), \tag{5a}$$

$$x^p(\delta(v)) + y^p(R_p^L(\delta(v))) + z_{v,p-1} + z_{vp} \equiv 0 \bmod 2 \quad (\forall v \subset V,\ p = 2,\dots,L), \tag{5b}$$

$$x_{ep} \in \{0,1,2\}; \quad y_{ep}, z_{vp} \in \{0,1\}. \tag{6}$$

The y variables do not appear in the objective function since the cost of servicing the required edges is fixed. The second component in the objective represents the cost of returning to the depot after servicing is completed. Constraints (1) state that each edge must be serviced exactly once, (2) are the time deadlines (after some simplification) and (3) ensure that there is only one phase transition at the end of each phase and one trip back to the depot after servicing the last required edge. Constraints (4a), (4b), and (4c) ensure that the route is connected in phase 1 and (4d), (4e) do the same for the other phases. Constraints (5) ensure that the vehicle leaves each vertex as many times as it enters: (5a) concerns phase 1 and (5b) the other phases. Finally, (6) are the integrality conditions and bounds. Note that LP software with provision for Special Ordered Sets can exploit the structure of (3) and possibly also (1).

The convex hull of solutions to (1)–(6) is a bounded polyhedron, i.e., a polytope. It will be denoted by **DC(G)**.

3. Valid inequalities

In this section, several classes of inequality are shown to be valid for **DC(G)**. Some of these have proved to be effective when used as cutting-planes, suggesting that they induce high-dimensional faces of **DC(G)**. However, the authors have not attempted to find conditions under which they induce facets, for three reasons: first, little is yet known even about multidimensional Knapsack polyhedra such as implied by constraints (2) and (6); second, **DC(G)** is not full-dimensional due to constraints (1) and (3); third, it is NP-Hard to calculate the dimension of **DC(G)** or even to determine if it is non-empty.

We first show a strong relationship between **DC(G)** and the polyhedra associated with various (ordinary) RPPs. For any integer B satisfying $1 \leq B \leq L$, define a graph $G^B(V^B, E^B)$, with $V^B = V \cup \{1^*\}$ and $E^B = E \cup \{1,1^*\} \cup \{\{v,1^*\}\ \forall v \in V\}$. The vertex $\{1^*\}$ may be thought of as a copy of the depot. Now consider an RPP defined on G^B, with arbitrary costs, in which the required edges are $R_1^B \cup \{1,1^*\}$. Associate a variable x'_e with each $e \in E^B$, representing the number of times e is traversed without servicing in an RPP solution. Consider the polyhedron **RPP(G^B)** representing the convex hull in $\mathscr{R}^{|E|+|V|+1}$ of incidence vectors corresponding to valid RPP solutions. From the results of Corberán and Sanchis (1994, 1996), Letchford (1996) and Letchford (1997), several classes of valid inequalities and facets can be defined for **RPP(G^B)**.

Theorem 1. *If any inequality of the form*

$$\sum_{e \in E} \alpha_e x'_e + \sum_{v \in V} \beta_v x'_{\{v,1^*\}} + \gamma x'_{\{1,1^*\}} \geq \delta$$

is valid for **RPP(G^B)**, *then the inequality*

$$\sum_{e \in E} \alpha_e x_1^B(e) + \sum_{e \in R_{B+1}^L} \alpha_e y_1^B + \sum_{v \in V} \beta_v z_{vB} \geq \delta$$

is valid for **DC(G)**.

Proof. Given any feasible solution to (1)–(6), construct a multigraph $G^*(V^*, E^*)$ as follows: Let $V^* = V^B$ and let E^* consist of the edge $\{1,1^*\}$, $x_1^B(e)$ copies of each edge $e \in E \setminus R$, $x_1^B(e) + 1$ copies of each edge in R^1, z_{vB} copies of edge $\{v,1^*\}$ and, for $2 \leq m \leq L$, $x_l^B(e) + y_1^{\min(m,B)}(e)$ copies of each $e \in R^m$. By construction, G^* is Eulerian and E^* contains at least one copy of each $e \in R_1^B$. But this means that G^* represents a feasible solution to the RPP defined on G^B. The inequality follows from the construction of G^* and the fact that $x'_{\{1,1^*\}} = 0$ in this feasible solution. □

We call the inequalities resulting from Theorem 1, *Strong Cumulative* (SC) inequalities. To explore the SC inequalities in detail would entail a thorough review of the above papers on the RPP, which space does not permit. However, three specific classes of SC inequality have proven to be particularly useful as cutting-planes. These are summarised below:

Corollary 2. *For some $1 \le B \le L$, let $S \subseteq V - \{1\}$ be such that $R_1^B(E(S)) \neq \emptyset$ and $R_1^B(\delta(S)) = \emptyset$. Then:*

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S))) + z^B(S) \ge 2 \tag{7}$$

is valid for **DC(G)**.

These are analogous to the Connectivity inequalities of the RPP (Corberán and Sanchis, 1994). Hence, we call them *Strong Cumulative Connectivity* (SCC) inequalities. Note that they generalise (4a).

Corollary 3. *If $1 \le B \le L$ and $S \subseteq V - \{1\}$, then:*

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S))) + z^B(S) \ge 1 \tag{8a}$$

is valid for **DC(G)** *when $|R_1^B(\delta(S))|$ is odd and*

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S))) + z^B(V - S) \ge 1 \tag{8b}$$

is valid for **DC(G)** *when $|R_1^B(\delta(S))|$ is zero or even.*

These are analogous to the R-Odd Cut inequalities of the RPP (Corberán and Sanchis, 1994). We call them *Strong Cumulative Parity* (SCP) inequalities.

Corollary 4. *Let B, P and W be integers with $1 \le B \le L$, $P \ge 1$, $W \ge 0$, $P + W \ge 3$ and odd. Let n_i $(i = 1,\dots,P)$ be integers greater than one. Partition V into sets M, N, V_j^i $(i = 1,\dots,P,\ j = 1,\dots,n_i)$. For convenience, for all i we identify V_j^i with M when $j = 0$ and with N when $j = n_i + 1$. If the partition is such that:*

- *for each edge $\{u,v\} \in R_1^B$, either $\{u\}$ and $\{v\}$ lie within the same $G(V_j^i)$, or one of $\{u\}$ and $\{v\}$ is in M and the other is in N;*
- *for $i = 1,\dots,P$, $j = 1,\dots,n_i$, either $R_1^B(E(V_j^i)) \neq \emptyset$, or $\{1\} \in V_j^i$, or both;*
- *there are W required edges in R_1^B with one end-vertex in M and the other in N;*

then the following inequality is valid for **DC(G)**:

$$\sum_{e\in E} \alpha_e x_e + \sum_{e\in R_{B+1}^L} \alpha_e y_e + \sum_{v\in V} \beta_v z_{vB} \ge 1 + \sum_{i=1} \frac{n_i + 1}{n_i - 1},$$

where α_e equals:

$$\alpha_e = \begin{cases} 1 & \text{if } i \in M,\ j \in N; \\ \dfrac{q-p}{n_r - 1} & \text{if } e \text{ connects } V_p^r \text{ to } V_q^r\ (p < q), \\ & \quad \text{either } p \neq 0 \text{ or } q \neq n_r + 1; \\ 1/(n_r - 1) + 1/(n_s - 1) + \left| \dfrac{p-1}{n_r - 1} - \dfrac{q-1}{n_s - 1} \right| & \text{if } i \in V_p^r,\ j \in V_q^s,\ r \neq s,\ 1 < p < n_r,\ 1 < q < n_s; \\ 0 & \text{otherwise}; \end{cases}$$

and $\beta_v = \alpha_e$, where $e = \{v,1\}$, the edge that would be created by connecting v to $\{1\}$.

These are analogous to the Path-Bridge inequalities of the RPP (Letchford, 1997), and are accordingly called *Strong Cumulative Path-Bridge* (SCPB) inequalities.

SCC, SCP and SCPB inequalities have proved to be essential to the running of the cutting-plane algorithm, since they produce the largest increases in the objective function value. Other Strong Cumulative inequalities can be obtained in a similar way, but we have not found them to be of practical use.

Unfortunately, **DC(G)** is much more complicated than implied by the results given so far. Consider the following results:

Theorem 5. *For some* $1 \le B < L$, *let* $S \subseteq V - \{1\}$ *be such that* $R_1^B(E(S) \cup \delta(S)) = \emptyset$, *but* $R_{B+1}^L(E(S)) \neq \emptyset$. *Then, for any* $e \in R^m(E(S))$, *with* $B < m \le L$,

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S))) + z^B(S) \ge 2y_1^m(e) \tag{9}$$

is valid for **DC(G)**.

Proof. If e is not serviced in phases 1 to B, the right-hand side becomes 0 and the inequality is trivially true. On the other hand, if e is serviced in phases 1 to B, the right-hand side becomes 2, which is valid since the vehicle must enter S at least once during phases 1 to B, and must then either leave S (possibly servicing an edge in R_{B+1}^L as it does so), or end phase B while still within S. □

Theorem 6. *If* $1 \le B < L$ *and* $S \subseteq V - \{1\}$, *then*:

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S) - F)) + z^B(S) \ge y_1^B(F) - |F| + 1 \tag{10a}$$

is valid for any $F \subseteq R_{B+1}^L(\delta(S))$ *such that* $|F \cup R_1^B(\delta(S))|$ *is odd and*

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S) - F)) + z^B(V - S) \ge y_1^B(F) - |F| + 1 \tag{10b}$$

is valid for any $F \subseteq R_{B+1}^L(\delta(S))$ *such that* $|F \cup R_1^B(\delta(S))|$ *is zero or even.*

Proof. If $|F| - 1$ or less of the edges in F are serviced in phases 1 to B, the right-hand side becomes zero or less and the inequality is trivially true. On the other hand, if all of the edges in F are serviced in phases 1 to B, the right-hand side becomes 1, which is valid since the vehicle must enter and leave S an even number of times. □

The inequalities (9), which generalise (4b), can be obtained by the following three step procedure: (i) temporarily fix y_{ep} to zero for $B < p \le m$, which is equivalent to moving e from R^m to R^B; (ii) invoke Corollary 2 for the new RPPDC instance thus formed; (iii) lift the resulting inequality back into the space of the original variables (for an introduction to lifting, see, e.g., Nemhauser and Wolsey, 1988). For this reason, we call them *Lifted Cumulative Connectivity* (LCC) inequalities. Similarly, the inequalities (10a) and (10b) can be obtained by fixing variables associated to the edges in F, invoking Corollary 3 and lifting; we call them *Lifted Cumulative Parity* (LCP) inequalities.

Other *Lifted Cumulative* inequalities can be found in a similar way, but only the LCC and LCP inequalities have proven to be useful in the cutting-plane algorithm. Even these do not tend to produce as large an increase in the objective function as the SC inequalities, although they are necessary to obtain feasibility.

The inequalities presented so far have all concerned parts of the route involving phase 1. It is possible to derive inequalities which concern only later phases, as illustrated below. The proofs are omitted because they are so similar to previous cases.

Theorem 7. *For any* $S \subseteq V$: $R_2^L(S) \neq \emptyset$, *and any* $e \in R^m(S)$, $m \ge 2$,

$$x_A^B(\delta(S)) + y_A^B(R_A^L(\delta(S))) + z^{A-1}(S) + z^B(S) \ge 2y_A^B(e) \tag{11}$$

is valid for any $2 \le A \le B \le m$.

Theorem 8. *For any $S \subseteq V$, $2 \le A \le B \le L$ and $F \subseteq R_A^L(\delta(S))$,*

$$x_A^B(\delta(S)) + y_A^B(R_A^L(\delta(S) - F)) + z^{A-1}(S) + z^B(S) \ge y_A^B(F) - |F| + 1 \quad (12a)$$

is valid when $|F|$ is odd and

$$x_A^B(\delta(S)) + y_A^B(R_A^L(\delta(S) - F)) + z^{A-1}(S) + z^B(V-S) \ge y_A^B(F) - |F| + 1 \quad (12b)$$

is valid when $|F|$ is even.

For obvious reasons, we call (11), (12a) and (12b) *Non-Cumulative Connectivity* (NC) and *Non-Cumulative Parity* (NP) inequalities, respectively. They tend to produce little increase, if any, in the objective function when used as cutting-planes. However, they are required in order to gain feasibility. Note that (11) generalise (4d).

With more effort, still more valid inequalities could be found. For example, constraints (1), (2) and (6) form a multidimensional Knapsack subproblem with upper bounds of 2 on the x variables and generalised upper bounds on the y variables. It might be possible to find useful inequalities which exploit this feature. As another example, the authors have constructed some special RPPDC instances in which all of the inequalities currently known are not sufficient to ensure that constraints (5) hold, even after integrality is obtained via branch-and-bound. Some new inequalities would be needed to cut off such infeasible integral solutions.

4. Separation routines

In order for the above inequalities to be used as cutting-planes, separation routines are required in order to detect when they are violated (see, e.g., Nemhauser and Wolsey, 1988). Fortunately, the various Connectivity and Parity inequalities presented in the previous section can be separated in polynomial time and there is a reasonable heuristic for SCPB inequalities. For the sake of brevity, we only present separation routines for the LCC, LCP and SCPB inequalities. Fairly minor adaptations of the first two yield routines for SCC, SCP, NC and NP inequalities.

First, the LCCs. Assume that B is fixed. Construct the following *B-graph*: copy $G(V,E)$, give each edge in R a weight of $x_1^B(e) + y_1^B(e)$ and each other edge a weight of $x_1^B(e)$. For each $v \in V$, add an extra edge $\{v,1\}$ weight of z_{vB}. Now let $K = \{v \in V$: v is an end-vertex of at least one $e \in R_1^B\} \cup \{1\}$. Delete $E(K)$ from the B-graph, move the vertices in K together until they coincide in a single 'supernode', then replace each resulting set of parallel edges, if any, with a single edge. The weight of each new edge thus formed should be equal to the sum of the weights of the corresponding parallel edges.

Call the resulting graph the *SB-graph* (S for 'shrunk'). The only edges in R_{B+1}^L which are suitable candidates to be e in (9) are those which are not incident on the supernode in the SB-graph. Now let $e = \{u,v\}$ be one such candidate. A minimum cut in the SB-graph such that $\{u\}$ and the supernode lie on opposite shores can be found by sending a maximum flow from $\{u\}$ to the supernode.

Lemma 9. *If the value of the cut is $\ge 2y_1^B(e)$, there is no LCC violated for the given B and e. If the value is $< 2y_1^B(e)$ and $\{u\}$ and $\{v\}$ are on the same shore, then an LCC is violated. If the value is $< 2y_1^B(e)$ and $\{u\}$ and $\{v\}$ are on opposite shores, then an LCP is violated with $F = \{e\}$.*

Proof. This follows from the construction of the graph and the fact that a LCP with $|F| = 1$ can be rewritten as:

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S))) + z^B(S) \ge 2y_1^B(F). \quad \square$$

The maximum possible number of times the maximum flow routine could need to be invoked is clearly $O(\min\{|V|, |R|\})$. As there are $O(|E|)$ edges in the SB-graph, we have:

Theorem 10. *If all LCPs are already satisfied, the separation problem for LCCs can be solved in the time taken to perform $O(L \cdot \min\{|V|,|R|\})$ maximum flow problems in a graph with $O(|V|)$ vertices and $O(|E|)$ edges.*

The routine can in fact be improved, due to the following result:

Lemma 11. *Suppose all LCPs are satisfied, B is fixed and an LCC is violated for some S and e. If f is an edge adjacent to e with $y_1^B(f) \geq y_1^B(e)$, then the LCC with e replaced by f is violated by at least as much.*

Proof. As e and f are adjacent, f is in either $R(E(S))$ or $R(\delta(S))$. If f is in $R(E(S))$, the result is immediate. Suppose f lies in $R(\delta(S))$. Because all LCPs are satisfied, we have

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S))) + z^B(S) \geq 2y_1^B(f) \geq 2y_1^B(e),$$

which contradicts the assumption that the LCC is violated for the given S and e. □

Thus only a few edges (the ones with locally maximal y values) need to be considered as candidates for e, which leads to fewer maximum flows being needed.

Now to consider the LCPs. First, note that (10a) and (10b) can be rewritten as:

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S) - F)) + \sum_{e \in F} (1 - y_1^B(e)) + z^B(S) \geq 1 \tag{13a}$$

and

$$x_1^B(\delta(S)) + y_1^B(R_{B+1}^L(\delta(S) - F)) + \sum_{e \in F} (1 - y_1^B(e)) + z^B(V - S) \geq 1, \tag{13b}$$

respectively.

We use the well-known edge-splitting strategy (see e.g., Padberg and Rao, 1982, Letchford, 1997). Create a copy of $G(V,E)$ and label vertices *even* unless they meet an odd number of edges in R_1^B, when they should be labelled *odd*. Give each $e \in R_{B+1}^L$ a weight of $x_1^B(e) + y_1^B(e)$, other edges a weight of $x_1^B(e)$. Now *split* each $e \in R_{B+1}^L$ into two new edges, called *halves*, by inserting a new *even* vertex in the middle. Let one half (the *normal* half) retain the previous weight, but let the other (the *flipped* half) have weight $1 + x_1^B(e) - y_1^B(e)$. Now add an extra odd vertex 1^* and reverse the label of the depot. From each $v \in V$, add an edge $\{v, 1^*\}$ of weight z_{iB}. Finally, reverse the label of any vertex which is adjacent to an odd number of flipped halves.

A cut in the resulting *split graph* will have weight equal to the left-hand-side of (13a) (respectively, (13b)) if $\{1\}$ and $\{1^*\}$ lie on the same (opposite) shores of the cut; moreover, the cut will be *odd* iff the edge-cutset contains an *odd* (*even*) number of flipped halves. This immediately yields the following result:

Lemma 12. *An LCP is violated iff there is an odd cutset in the split graph with a weight less than* 1. *If* $\{1\}$ *and* $\{1^*\}$ *are on the same shore of the cut, it is of the form* (13a); *but if they are on opposite shores, it is of the form* (13b). *The set F consists of those $e \in R_{B+1}^L$ whose flipped halves lie in the edge-cutset.*

The best current algorithm for finding a minimum weight odd-cut is that of Padberg and Rao (1982). It entails the solution of N_{ODD} maximum flow problems, where N_{ODD} is the number of odd vertices. Now, the split graph contains $O(\max\{|V|,|R|\})$ vertices, $O(|E|)$ edges and $O(|R|)$ odd vertices. Therefore:

Theorem 13. *The separation problem for LCPs can be solved in the time taken to perform $O(L \cdot |R|)$ maximum flow problems in graphs with $O(\max\{|V|,|R|)$ vertices and $O(|E|)$ edges.*

Finally, we sketch our heuristic routine for generating useful SCPB inequalities. In Applegate et al. (1994)

and Fleischer and Tardos (1996), efficient heuristic separation routines are given to identify violated *Comb* inequalities, which, in the terminology of the present paper, are analogous to SCPB inequalities in which $n_i = 2$ for all i, and $W = 0$. These routines involve setting up a data structure which can compactly represent all sets S of vertices in a weighted graph such that $\delta(S)$ is a minimum cut in that graph. This is then followed by a search for pairs of vertex sets S_1, S_2 with the property that S_1, S_2 and $S_1 \cup S_2$ form a minimum cut. Such pairs are good candidates for sets V_1^i and V_2^i in a path.

It is not hard to adapt these routines, to find violated Path-Bridge inequalities with $n_i = 2$ for all i, in the context of the ordinary RPP. Moreover, if such an inequality is found, one can attempt, for all i, to increase n_i (i.e., to 'extend' the length of the paths). This can be done by removing suitable sets of vertices from M or N, if possible, and making them new V_j^i, until neither M nor N contain any suitable sets. A similar procedure is described in Clochard and Naddef (1993).

We apply this to the RPPDC as follows: First, we solve a given RPPDC instance as if it were an ordinary RPP, identifying violated Connectivity, R-Odd Cut and Path-Bridge inequalities. Then, we convert the binding inequalities into their Strong Cumulative counterparts, and use them when solving the RPPDC proper. Although this is a straightforward approach, it appears to work sufficiently well.

5. Computational experiments

In this section, some computational results are given for a number of RPPDC problem instances. The optimisation strategy used is the dual cutting-plane method (see Nemhauser and Wolsey, 1988) in which an initial LP relaxation is solved and then violated inequalities are identified and added to the LP as cutting-planes. Each time an inequality is added, the LP is resolved using the dual simplex method. When no more violated inequalities can be found, branch-and-bound is invoked to obtain integrality.

It may happen that the resulting integer solution violates one or more known inequalities. If this happens, the inequalities are appended to the LP and the cutting-plane procedure continues again, followed by branch-and-bound once more, and so on. Such restarts, which were done manually, were necessary since branch-and-cut software was unavailable at the time of writing. The number of restarts needed ranged from zero to about fifty in the problems we tested, so branch-and-cut software would speed up the algorithm significantly.

The LP solver used is CPLEX version 3.0. A C routine for constructing a Gomory–Hu cut tree was kindly given to the authors by Georg Skorobohatyj at Z.I.B. in Berlin. Additional C routines for constructing the necessary graphs and forming inequalities were written by the authors. The hardware used is a Hewlett Packard HP-9000 workstation.

As mentioned in the previous section, the initial relaxation is constructed by ignoring the time deadlines initially, solving the RPPDC instance as an ordinary RPP, then transferring the binding inequalities over to the RPPDC. The SCC, SCP, LCC, LCP, NC and NP separation routines are then invoked in cyclic order until a violated inequality is found.

The test problems are adapted from Corberán and Sanchis (1994). Corberán and Sanchis present 26 RPP

Table 1
RPP problem instances

Instance	$\lvert V \rvert$	$\lvert E \rvert$	$\lvert R \rvert$	Comps	Con	R-Odd	PBI	Optimum
I4	17	35	22	3	1	7	3	29
I14	28	79	31	6	5	21	2	57
I16	31	94	34	7	6	18	2	64
I20	50	98	63	7	9	20	2	116
I21	49	110	67	6	4	22	5	78

Table 2
Results for single deadline class

Instance	T^1	Con	Par	LP	B&B	Optimum
I4	105	0	3	29.67	20	33
I14	260	0	0	57.00	15	59
I16	263	0	3	64.43	27	67
I20	522	0	4	116.49	28	118
I21	490	0	2	75.67	180	84

instances, but 15 of these can be solved using only Connectivity and R-Odd Cut inequalities, and a further 6 can be solved by adding only a single Path-Bridge inequality. The authors decided to base the RPPDC instances on the remaining 5, all of which require at least 2 Path-Bridge inequalities to be solved: I4, I14, I16, I20 and I21. I21 is especially difficult in that it cannot be solved by Connectivity, R-Odd and Path-Bridge inequalities alone. When the authors solved it, 13 branch-and-bound nodes were required.

Table 1 displays the problem characteristics. The column labelled 'Comps' shows the number of connected components in the graph obtained when edges in $E \setminus R$ are removed. The next three columns show the number of binding Connectivity, R-Odd Cut and Path-Bridge inequalities in the final LP relaxation when solved as an ordinary RPP, and the last column shows the optimum RPP cost.

We solved two deadline versions of each problem: one with $L = 1$ and one with $L = 2$. In each case, we set t_e equal to c_e and s_e equal to $3c_e/2$ rounded down to the nearest number. Such a high correlation between times and costs is often encountered in practice. The versions with $L = 1$ were formed by making the deadline just tight enough to make the ordinary RPP solution invalid. The versions with $L = 2$ were formed by a slightly more complicated procedure as follows:

Partition R into equivalence classes, putting two edges in the same class if they have the same cost. List the classes in decreasing order of cost and set $R^1 = \emptyset$ and $R^2 = R$. Then do the following until $|R^1| \geq |R^2|$: remove a class C from the head of the list, set $R^1 := R^1 \cup C$ and $R^2 := R^2 \setminus C$. This led to $|R^1|$ values of 11, 20, 19, 40 and 38, respectively, for the five problems. A 'plausible' solution was then formed by hand, and the deadlines set accordingly.

The computational results are given in Tables 2 and 3. For each problem the following is listed: the time deadline(s), the number of cutting-planes of each type generated in addition to those in the initial relaxation, the cost of the final LP relaxation, the number of branch-and-bound nodes required to solve the final relaxation to optimality and the cost of the optimal solution. It will be seen that all ten instances could be solved to optimality by the cutting-plane method.

Due to the manual restarts, it is not too meaningful to talk in terms of computation time. It is worth mentioning, however, that the time taken by the LP solver and the separation routines, combined, never exceeded 3 min for any problem; also, each single invocation of branch-and-bound took less than 20 s. The authors are confident that each of the problems could be solved within a few minutes if branch-and-cut software was available.

Table 3
Results for two deadline classes

Instance	T^1	T^2	Con	Par	LP	B&B	Optimum
I4	92	110	28	50	30.35	25	33
I14	236	321	33	42	57.23	63	62
I16	229	268	29	27	67.65	147	72
I20	498	624	37	52	116.17	131	118
I21	468	500	56	81	75.84	2657	84

6. Conclusion

It is quite possible to solve realistic RPPDC examples to optimality using the specialised cutting-planes presented in this paper. This is in large part due to the way in which the special structure of the time deadlines has been exploited.

There are a number of areas in which progress could be made: Finding further valid inequalities; devising new separation routines; adapting the formulation to node-routing, multiple vehicles or networks containing directed edges; devising efficient heuristics for very large problems; etc. The authors have already produced some valid inequalities and separation routines for multi-vehicle problems in which $L = 1$ (Letchford and Eglese, 1996), but further research is necessary to deal with problems of greater generality.

Acknowledgements

Thanks are due to the three anonymous referees for their helpful comments.

References

Applegate, D., Bixby, R.E., Chvátal, V., Cook, W., 1994. Finding cuts in the TSP. Working paper, Bell Communications Research, Morristown, New Jersey, 07690, USA.

Clochard, J.M., Naddef, D., 1993. Using path inequalities in a branch-and-cut code for the symmetric travelling salesman problem. In: Rinaldi, G., Wolsey, L. (Eds.), Proc. of the 3rd Int. IPCO Conference. Centro Ettore Majorana, Erice, Italy.

Corberán, A., Sanchis, J.M., 1994. A polyhedral approach to the rural postman problem. European Journal of Operational Research 79, 95–114.

Corberán, A., Sanchis, J.M., 1996. The general routing problem polyhedron: Facets from the RPP and GTSP polyhedra. To appear in European Journal of Operational Research.

Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F., 1995. Time-constrained routing and scheduling. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (Eds.), Network Routing, Handbooks on Operations Management and Management Science, vol. 8. Elsevier, Amsterdam.

Dumas, Y., Desrosiers, J., Gélinas, J., Solomon, M., 1995. An optimal algorithm for the travelling salesman problem with time windows. Operations Research 43, 367–371.

Eiselt, H.A., Gendreau, M., Laporte, G., 1995a. Arc-routing problems, Part 1: the Chinese postman problem. Operations Research 43, 231–242.

Eiselt, H.A., Gendreau, M., Laporte, G., 1995b. Arc-routing problems, Part 2: the rural postman problem. Operations Research 43, 399–414.

Fleischer, L., Tardos, E., 1996. Separating maximally violated comb inequalities in planar graphs. Technical report TR1150. School of Ops. Res. and Ind. Eng., Cornell University.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractibility: A Guide to the Theory of NP-Completeness. Freeman, San Francisco.

Kohl, N., Madsen, O., 1995. An optimisation algorithm for the vehicle routing problem with time windows based on Lagrangean relaxation. Working paper, Institute of Mathematical Modelling, Technical University of Denmark.

Lenstra, J.K., Rinnooy Kan, A.H.G., 1976. On general routing problems. Networks 6, 273–280.

Letchford, A.N., 1996. Further inequalities for the general routing problem. To appear in the European Journal of Operational Research.

Letchford, A.N., 1997. New inequalities for the general routing problem. European Journal of Operational Research 96, 317–322.

Letchford, A.N., Eglese, R.W., 1996. New valid inequalities for vehicle routing problems. Working paper, Dept. of Management Science, Lancaster Unversity.

Li, L.Y.O., Eglese, R.W., 1996. An interactive algorithm for winter gritting. Journal of the Operational Research Society 47, 217–228.

Nemhauser, G.L., Wolsey, L.A., 1988. Integer and Combinatorial Optimisation. Wiley, New York.

Nygard, K.E., Greenberg, P., Bolkan, W.E., Swenson, J.E., 1988. Generalised assignment methods for the deadline vehicle routing problem. In: Golden, B.L., Assad, A.A. (Eds.), Vehicle Routing: Methods and Studies. North-Holland, Amsterdam.

Orloff, C.S., 1974. A fundamental problem in vehicle routing. Networks 4, 35–64.

Padberg, M., Rao, M.R., 1982. Odd minimum cut-sets and b-matchings. Mathematics of Operations Research 7, 67–80.

Stephenson, M., 1996. Devising skeletal routes for a parcel delivery company. Ph.D. thesis, Department of Man. Science, Lancaster University.

Thangiah, S.R., Osman, I., Vinayagamoorthy, R., Sun, T., 1994. Algorithms for the vehicle routing problem with time deadlines. American Journal of Mathematical and Management Sciences 13, 323–384.