Journal of the Operational Research Society (2007) 58, 1642–1651 0160-5682/07 $30.00

www.palgrave-journals.com/jors

A branch-and-cut algorithm for the capacitated open vehicle routing problem

AN Letchford[1]*, J Lysgaard[2] and RW Eglese[1]

[1]Lancaster University, Lancaster, UK; and [2]Aarhus School of Business, Denmark

In *open* vehicle routing problems, the vehicles are not required to return to the depot after completing service. In this paper, we present the first exact optimization algorithm for the open version of the well-known *capacitated vehicle routing problem* (CVRP). The algorithm is based on branch-and-cut. We show that, even though the open CVRP initially looks like a minor variation of the standard CVRP, the integer programming formulation and cutting planes need to be modified in subtle ways. Computational results are given for several standard test instances, which enables us for the first time to assess the quality of existing heuristic methods, and to compare the relative difficulty of open and closed versions of the same problem.

Journal of the Operational Research Society (2007) **58,** 1642–1651. doi:10.1057/palgrave.jors.2602345
Published online 13 December 2006

Keywords: vehicle routing; branch-and-cut; separation

Introduction

In the classical version of vehicle routing problems (VRPs), the vehicles are required to return to the depot after completing service (eg, Toth and Vigo, 2002). In *open* VRPs, however, the vehicles need not do so. As a result, the vehicle routes are not closed paths but open ones, starting at the depot and ending at one of the customers. Figure 1, which shows the optimal solutions to both the open and closed version for the same input data (all customers have unit demands and the vehicle capacity is five units), illustrates the fact that, in general, the optimal solution for the open version of a VRP can be quite different from that for the closed version. (Throughout this paper, the depot is represented by a white square and the customers by black circles.)

At first sight, having open routes instead of closed ones looks like a minor modification. Indeed, if travel costs are asymmetric, there is essentially no difference between the open and closed versions: to transform the open version into the closed one, it suffices to set the cost to zero for travelling from any customer to the depot. However, if travel costs are symmetric, things are more subtle. Indeed, we prove in the next section that, somewhat surprisingly, the open version turns out to be *more general* than the closed one, in the sense that any closed VRP on n customers can be transformed into an open VRP on n customers, but there is no transformation in the reverse direction.

Moreover, there are many practical applications in which open VRPs naturally arise. This happens, for example, when a company does not own a vehicle fleet and all its deliveries from a central depot are undertaken by hired vehicles that are not obliged to return to the depot. In such situations, the cost of the distribution may be proportional to the distance travelled while loaded. A practical case study of this type is described in Tarantilis *et al* (2004, 2005). The same model can also be used for pick-ups, where vehicles start empty at any customer and must pick up the demands of each customer on their route and deliver them to the depot.

There are also applications where the vehicles start at the depot, deliver to a set of customers and then are required to visit the customers in reverse order, picking up items that are required to be backhauled to the depot. If, for each customer, the pick-up demand is no larger than the delivery demand, then an open VRP model can be used. An application of this type for an air express courier is mentioned by Schrage (1981) in an early paper describing features of practical routing problems.

Two further areas of application are described by Fu *et al* (2005). The first involves the planning of train services, starting or ending at the Channel Tunnel. The second involves planning a set of school bus routes where in the morning pupils are picked up at various locations and brought to school, and in the afternoon, the routes are reversed to take pupils home. Bodin *et al* (1983) includes a description of a problem of express airmail distribution in the USA, that is essentially an open pick-up and delivery VRP with capacity constraints and time windows.

Open VRPs are easily seen to be strongly $\mathcal{NP}$-hard by reduction from the Hamiltonian path problem. Research on open VRPs has therefore up to now concentrated on devising effective heuristics for solving them. For the version involving only capacity constraints, Sariklis and Powell

*Correspondence: *AN Letchford, Department of Management Science, Lancaster University, Lancaster, LA1 4YW UK.*
E-mail: A.N.Letchford@lancaster.ac.uk

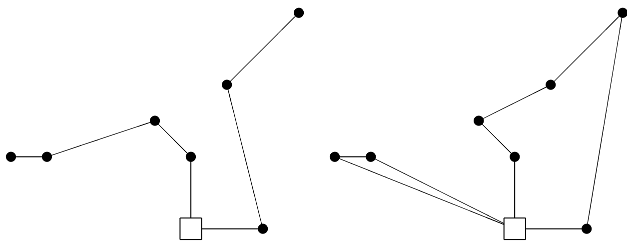

Figure 1 Open *versus* closed routes with different clusterings of customers.

(2000) presented a two-phase heuristic involving minimum spanning trees, Tarantilis *et al* (2004) present a population-based heuristic, and Tarantilis *et al* (2005) present a heuristic of the threshold-accepting type. For a more general variant involving both capacity and route-length constraints, Brandão (2004) and Fu *et al* (2005, 2006) describe tabu search heuristics, Li *et al* (2006) present a record-to-record travel heuristic, and Pisinger and Ropke (2006) present an adaptive neighbourhood search heuristic. Heuristics have also been devised for open VRPs with other kinds of constraints; see, for example, Repoussis *et al* (2006) and Aksen *et al* (2006).

In this paper, we present the first exact optimization algorithm for the capacitated open vehicle routing problem (COVRP), which is defined as follows. A complete undirected graph $G=(V, E)$ is given, with $V=\{0,\ldots,n\}$. Vertex 0 represents the depot, the other vertices represent customers. The cost of travel from vertex i to vertex j is denoted by c_{ij}, and we assume costs are symmetric, so $c_{ij}=c_{ji}$. A fleet of K identical vehicles, each of capacity $Q>0$, is given. Each customer i has a demand q_i, with $0<q_i\leqslant Q$. Each customer must be serviced by a single vehicle and no vehicle may serve a set of customers whose total demand exceeds its capacity. Each vehicle route must start at the depot and end at the last customer it serves. The objective is to define a set of vehicle routes that minimizes the total costs.

As we will show, our algorithm is capable of solving small- to medium-size instances to optimality, and providing useful lower bounds for larger instances. It can also be easily adapted to handle some other variants of the COVRP, such as variants with a free vehicle fleet size, or with a fixed cost associated with the use of a vehicle.

The structure of the remainder of the paper is as follows. In the next section, we give an integer programming formulation of the COVRP and present some valid inequalities. It will be seen that more complex inequalities are needed for the open version than for the closed version. Then, in the following section, we describe the ingredients of our branch-and-cut algorithm. Extensive computational results are given in the following section, which enable us for the first time to assess the quality of existing heuristic methods, and to compare the relative difficulty of open and closed versions of the same problem. Some concluding remarks are given in the final section.

Formulation and valid inequalities

Formulation

The COVRP is clearly a special case of the asymmetric CVRP (ACVRP), in which, for any $i, j \in V$, c_{ij} is permitted to be different from c_{ji}. Hence, it would be possible to use any integer programming formulation of the ACVRP (eg, that of Fischetti *et al*, 1994) to solve the COVRP. However, this would mean that effectively we were treating each (undirected) edge as two (directed) arcs, which would lead to a formulation of the COVRP with twice as many variables as our formulation of the CVRP. This seems unnecessary, given that, in the COVRP, $c_{ij}=c_{ji}$ when i and j are customers.

A more parsimonious formulation of the COVRP can be obtained by modifying the standard formulation of the CVRP. To explain this clearly, it is helpful to recall the following details of the CVRP formulation.

Let $V_c = V\backslash\{0\}$ denote the set of customers. Given a set $S \subseteq V_c$, let $q(S)$ denote $\sum_{i\in S} q_i$, $\delta(S)$ denote the set of edges in G with exactly one end-vertex in S, $E(S)$ denote the set of edges in G with both end-vertices in S, and $k(S)$ denote $\lceil q(S)/Q\rceil$. Obviously, $k(S)$ is a lower bound on the minimum number of vehicles required to serve the customers in S. Let x_{ij} represent the number of times a vehicle travels between vertices i and j. (Because the problem is undirected, x_{ij} and x_{ji} represent the same variable.) Finally, given an arbitrary $F \subseteq E$, $x(F)$ will denote $\sum_{e\in F} x_e$. Then the standard (so-called *two-index*) formulation of the CVRP is (Laporte *et al*, 1985):

$$\text{Minimize} \quad \sum_{e\in E} c_e x_e$$

Subject to:

$$x(\delta(\{i\}))=2 \quad (i=1,\ldots,n) \tag{1}$$

$$x(\delta(S))\geqslant 2k(S) \quad (S\subseteq V_c, |S|\geqslant 2) \tag{2}$$

$$x(\delta(\{0\}))=2K \tag{3}$$

$$x_{ij}\in\{0,1\} \qquad (1\leqslant i<j\leqslant n) \tag{4}$$

$$x_{0i}\in\{0,1,2\} \qquad (i=1,\ldots,n) \tag{5}$$

The *degree equations* (1) ensure that customers are visited exactly once. *The rounded capacity inequalities* (2) impose the vehicle capacity restrictions and also ensure that the routes are connected. They can be re-written, using the degree equations, in the alternative form

$$x(E(S))\leqslant |S|-k(S) \tag{6}$$

Equation (3) ensures that exactly K vehicles are used. Finally, constraints (4) and (5) are the integrality conditions. Note that the variables x_{0i} are permitted to take the value 2, to allow routes in which a vehicle serves a single customer.

To adapt this formulation to the COVRP, we simply treat each edge incident on the depot as a pair of directed arcs, as

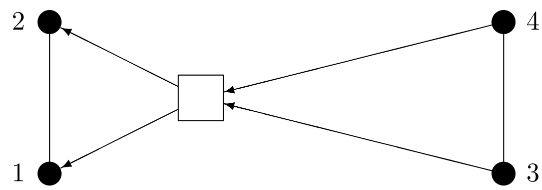

Figure 2 Invalid integer solution.

follows. For each $i \in V_c$, instead of defining the undirected variable x_{0i}, we define the binary variable y_{0i}, which takes the value 1 if and only if a vehicle travels directly from the depot to i, and the variable y_{i0}, which takes the value 1 if and only if a vehicle travels directly from i to the depot. We also use the notation $y^-(S) = \sum_{i \in S} y_{0i}$ and $y^+(S) = \sum_{i \in S} y_{i0}$. Finally, for any $S \subseteq V_c$ we use the notation $\bar{S} = V_c \backslash S$ and $\bar{\delta}(S) = \{\{i, j\} : i \in S, j \in \bar{S}\}$.

If we adapt the above formulation to the COVRP in a straightforward manner, we obtain the following integer program:

$$\text{Minimize} \quad \sum_{e \in E(V_c)} c_e x_e + \sum_{i \in V_c} c_{0i} y_{0i}$$

$$\text{Subject to:} \quad x(\bar{\delta}(i)) + y_{0i} + y_{i0} = 2 \quad (i = 1, \ldots, n) \quad (7)$$

$$x(\bar{\delta}(S)) + y^-(S) + y^+(S) \geqslant 2k(S) \quad (S \subseteq V_c, |S| \geqslant 2) \quad (8)$$

$$y^-(V_c) = K \quad (9)$$

$$y^+(V_c) = K \quad (10)$$

$$x_{ij} \in \{0, 1\} \quad (1 \leqslant i < j \leqslant n) \quad (11)$$

$$y_{0i}, y_{i0} \in \{0, 1\} \quad (i = 1, \ldots, n) \quad (12)$$

The constraints (7), (8) are straighforward adaptations of the degree equations and rounded capacity inequalities, respectively. The inequalities (8) can again be re-written in the form $x(E(S)) \leqslant |S| - k(S)$. The constraints (9) and (10) ensure that exactly K vehicles leave and enter the depot. Finally, constraints (11) and (12) ensure that all variables are binary. (There is no longer any need to allow any variables to take the value 2.)

Perhaps surprisingly, the above integer program does not represent a valid formulation for the COVRP. Figure 2 shows a solution to the above integer program for a small COVRP instance with $n = 4$, which does not represent a valid solution to the COVRP.

To prevent invalid solutions of this kind, it is necessary to add the following constraints to the formulation:

$$x(\bar{\delta}(S)) + y^+(S) \geqslant y^-(S) \quad (S \subseteq V_c, |S| \geqslant 2) \quad (13)$$

We call these constraints *balancing* inequalities. The fact that they are valid, and sufficient to ensure feasibility, follows from the conditions of Ford and Fulkerson (1962) for a mixed graph to be Eulerian. (Some similar inequalities were introduced by Nobert and Picard (1996), in the context of the so-called *Mixed Chinese Postman Problem*.)

The invalid solution above, for example, violates the balancing inequality with $S = \{1, 2\}$, which takes the form

$$x_{13} + x_{14} + x_{23} + x_{24} + y_{10} + y_{20} \geqslant y_{01} + y_{02}$$

It turns out that, once the balancing inequalities have been added to the formulation, Equation (9) is redundant.

Note that there are an exponential number of balancing inequalities. The need for balancing inequalities, which have no counterpart for the standard CVRP, suggests that the COVRP is a more complex problem than the CVRP. This is confirmed by the following definition and proposition.

Definition 1 The *partially asymmetric* CVRP (PACVRP) is the generalization of the CVRP in which the cost of travel c_{0i} is permitted to be different from c_{i0}.

Obviously, the PACVRP is intermediate in generality between the CVRP and ACVRP. The following result is less obvious.

Proposition 1 *The COVRP and the PACVRP are equivalent.*

Proof Any COVRP instance is clearly a PACVRP instance. Now, suppose we are given a PACVRP instance on n vertices, with symmetric travel costs c_{ij} for all $\{i, j\} \in E(V_c)$ and asymmetric travel costs c_{0i}, c_{i0} for all $i \in V_c$. Now let M be an arbitrary constant. Owing to the presence of the degree equations (7), (9) and (10), the optimal solution to the PACVRP is unchanged if we replace the original travel costs c_{ij} with modifed costs c'_{ij} defined as follows:

- for all $\{i, j\} \in E(V_c)$, $c'_{ij} = c_{ij} - c_{i0} - c_{j0} + M$,
- for all customers i, $c'_{0i} = c_{0i} - c_{i0} + M$ and $c'_{i0} = 0$.

If we choose M appropriately, the transformed costs c'_{ij} will be non-negative. Since the costs c'_{i0} are now zero, we have a COVRP instance. □

The algorithm we propose in this paper can therefore be used to solve instances of the PACVRP.

We remark that an alternative integer programming formulation for the COVRP can be obtained by eliminating the variables y_{i0}, which do not appear in the objective function, via Equation (7). Although the resulting formulation has n fewer variables, it is harder to understand and analyse and, more importantly, has a higher density (proportion of non-zeroes), which is unattractive from a computational point of view. For these reasons, we prefer to use the original formulation.

In the following subsections, we examine various valid inequalities for the integer polyhedron associated with the above formulation.

Symmetric inequalities

From a polyhedral point of view, the COVRP is similar to the CVRP. The following proposition, which is trivial to prove, shows that any valid inequality for the CVRP yields a valid inequality for the COVRP.

Proposition 2 *Let* $\sum_{0\leqslant i<j\leqslant n} \alpha_{ij} x_{ij} \leqslant \beta$ *be valid for the CVRP. Then* $\sum_{1\leqslant i<j\leqslant n} \alpha_{ij} x_{ij} + \sum_{i=1}^{n} \alpha_{0i}(y_{0i} + y_{i0}) \leqslant \beta$ *is valid for the COVRP.*

We call inequalities obtained in this way *symmetric*. A simple example of a class of symmetric inequalities is given by the inequalities (8), which are clearly the symmetric version of the rounded capacity inequalities (2). Other valid inequalities for the CVRP include, for example, *the homogeneous multistar* and *partial multistar*, *generalized large multistar*, *framed capacity*, *strengthened comb*, and *hypotour* inequalities. See Augerat (1995), Letchford *et al* (2002), Lysgaard *et al* (2004) and Naddef and Rinaldi (2002) for details. From Proposition 2, these all have valid counterparts for the COVRP.

Asymmetric inequalities

We say that a valid inequality $\alpha x + \beta y \geqslant \gamma$ for the COVRP is *asymmetric* if there exists at least one $i \in V_c$ such that $\beta_{0i} \neq \beta_{i0}$. The existence and necessity of the balancing inequalities shows that there exist non-redundant asymmetric inequalities. This should be expected, since the COVRP is a generalization of the CVRP.

Using the degree equations, it is possible to write the balancing inequality in a variety of forms. In particular, the balancing inequality for S is equivalent to $x(\bar{\delta}(\bar{S})) + y^-(\bar{S}) \geqslant y^+(\bar{S})$. Therefore, there is no need for a 'reversed' form of the balancing inequality, of the form $x(\bar{\delta}(S)) + y^-(S) \geqslant y^+(S)$, since this is equivalent to the balancing inequality on $\bar{S}$.

Unfortunately, the addition of all symmetric inequalities to the bounds, degree equations and balancing inequalities still does not give a complete description of the COVRP polyhedron. Suppose $n = 6$, $Q = 5$ and $q_i = 1$ for $i = 1, \ldots, 6$, and consider the fractional point displayed in Figure 3. (The dotted lines represent edges whose variables have value 1/2.) It is easy to check that it satisfies the bounds, degree equations and balancing inequalities. Moreover, it satisfies all symmetric inequalities. To see this, note that, if we replace the directed arcs with undirected edges, the resulting fractional point for the CVRP is a convex combination of the two integral points displayed in Figure 4.

The fractional point displayed in Figure 3 can be cut off by the inequality $x_{12} + x_{15} + x_{25} + x_{56} + y_{01} + y_{20} \leqslant 4$. This inequality, which is easily seen to be valid for the COVRP, is a special case of a class of inequalities which we call *mixed strengthened comb* inequalities. These inequalities are presented in the following theorem.

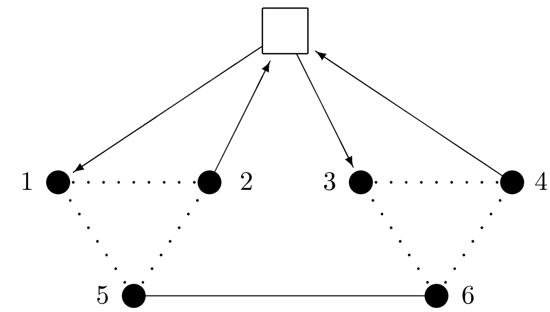

Figure 3 Fractional point satisfying all balancing and symmetric inequalities.

Theorem 1 *Let* $H \subset V_c$ *(the handle) and* $T_1, \ldots, T_t \subset V$ *(the teeth) be such that:*

- *every tooth properly intersects with the handle, that is* $T_i \cap H$ *and* $T_i \backslash H$ *are non-empty for all* i;
- *if any pair of teeth intersect, then either all vertices in the intersection lie in the handle or all lie outside, that is, for* $1 \leqslant i < j \leqslant t$, *either* $T_i \cap T_j \subset H$ *or* $T_i \cap T_j \cap H = \emptyset$.

Moreover, let the index set $\{1, \ldots, t\}$ *be partitioned into sets* $\mathscr{N}, \mathscr{D}, \mathscr{S}, \mathscr{R}$ *such that* $0 \notin T_i$ *for all* $i \in \mathscr{N}$ *and* $0 \in T_i$ *for all* $i \in \mathscr{D} \cup \mathscr{S} \cup \mathscr{R}$. *We call* T_i *a normal tooth if* $i \in \mathscr{N} \cup \mathscr{D}$, *a sending tooth if* $i \in \mathscr{S}$ *and a receiving tooth if* $i \in \mathscr{R}$. *Now, for any tooth* T_i, *we define*:

$$\gamma(T_i) = \begin{cases} k(T_i) + k(T_i \cap H) + k(T i \backslash H) & \text{if } i \in \mathscr{N} \\ k(V \backslash T_i) + k(T_i \cap H) + k(V \backslash (T_i \backslash H)) & \text{if } i \in \mathscr{D} \\ k(T_i \cap H) + 2 & \text{if } i \in \mathscr{S} \cup \mathscr{R} \end{cases}$$

Then, if $\sum_{i=1}^{t} \gamma(T_i)$ *is odd, the mixed comb inequality*

$$x(E(H)) + \sum_{i\in\mathscr{N}} x(E(T_i)) + \sum_{i\in\mathscr{D}\cup\mathscr{S}\cup\mathscr{R}} x(E(T_i \backslash \{0\})) + \sum_{i\in\mathscr{D}\cup\mathscr{S}} y^+(T_i \backslash \{0\}) + \sum_{i\in\mathscr{D}\cup\mathscr{R}} y^-(T_i \backslash \{0\}) \leqslant |H| + \sum_{i=1}^{t} |T_i| + |\mathscr{D}|(K-1) - \left\lceil \sum_{i=1}^{t} \gamma(T_i) \Big/ 2 \right\rceil \quad (14)$$

is valid for the COVRP.

The proof of this theorem is in Appendix A.

The mixed strengthened comb inequalities reduce to ordinary strengthened comb inequalities (Lysgaard *et al*, 2004) when there are no sending and receiving teeth. The fractional

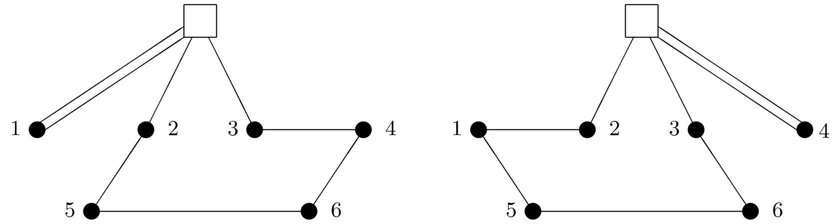

Figure 4 Two feasible CVRP solutions.

point displayed in Figure 3 violates the mixed comb inequality with handle $H = \{1, 2, 5\}$, normal tooth $\{5, 6\}$, receiving tooth $\{0, 1\}$ and sending tooth $\{0, 2\}$.

For many instances of the CVRP or COVRP, the number of vehicles K is fixed at the minimum possible, which often equals $k(V_c)$. In such a case, a lower bound on the amount which must be loaded onto any vehicle in any feasible solution is $q_{\min} = q(V_c) - Q(K - 1) \geqslant 0$. It is easy to show that it is never worthwhile having a vertex set as a sending or receiving tooth unless the total demand of that set is at least $q_{\min}$, and that it is not worthwhile having a customer set T as a normal tooth if $k(T) = k(T \cap H) + k(T \backslash H)$.

The branch-and-cut algorithm

Our implementation of the branch-and-cut algorithm for the COVRP follows to a large extent the implementation of our branch-and-cut algorithm for the CVRP, which was described in detail in Lysgaard *et al* (2004). In particular, the ingredients for *branching*, *node selection*, *cut pool management*, *and basis reconstruction* are implemented as in Lysgaard *et al* (2004).

As such, we focus here on those parts of our implementation for the COVRP that have no direct counterpart in our implementation for the CVRP. In the next three subsections, we describe our separation algorithms for symmetric, balancing, and mixed strengthened comb inequalities, respectively, and after that we describe the overall separation strategy.

We let (x^*, y^*) denote the current LP solution satisfying (7), (9), (10), and the bounds implied by (11), (12).

Separation of symmetric inequalities

As noted in Proposition 2, valid inequalities for the COVRP can be obtained by a simple reformulation of valid inequalities for the CVRP. This allows us to exploit the fact that effective separation algorithms have been developed for several classes of inequalities for the CVRP. In particular, in Letchford *et al* (2002) and Lysgaard *et al* (2004), we described effective separation algorithms for rounded capacity, homogeneous multistar and partial multistar, generalized large multistar, framed capacity, strengthened comb, and hypotour inequalities. These algorithms have been made publicly available (Lysgaard, 2003). We use these separation routines in our branch-and-cut algorithm, but with a modification to the strengthened comb separation routine to cope with the existence of sending and receiving teeth.

For details of these routines and the issue of sparse representations of the inequalities, we refer to Letchford *et al* (2002), Lysgaard (2003) and Lysgaard *et al* (2004).

Separation of balancing inequalities

The balancing inequalities can easily be separated in polynomial time by reduction to a max-flow/min-cut problem. We add $y^-(\bar{S})$ to both sides of (13) to obtain:

$$x(\overline{\delta}(S)) + y^+(S) + y^-(\overline{S}) \geqslant K \tag{15}$$

Thus, a balancing inequality is violated for a set S if and only if the left-hand side of (15), computed with respect to (x^*, y^*), is less than K. So, construct a directed graph $\hat{G} = (\hat{V}, \hat{A})$ in the following way. The vertex set $\hat{V}$ is set to $V \cup \{n+1\}$. For each $\{i, j\} \in E(V_c)$ such that $x^*_{ij} > 0$, insert two arcs (i, j) and (j, i) into $\hat{A}$ and give them a capacity of x^*_{ij}. For each $i \in V_c$, such that $y^*_{0i} > 0$ (respectively, $y^*_{i0} > 0$), insert the arc $(0, i)$ (respectively, $(i, n+1)$) into $\hat{A}$ and give it a capacity of y^*_{0i} (respectively, y^*_{i0}). Then, send a maximum flow in $\hat{G}$ from 0 to $n+1$, to find a minimum capacity $(0, n+1)$-cut. If the cut has a capacity less than K, a balancing inequality is violated, and the desired set S is given by the customer set on the same shore of the cut as 0.

It often happens that the subgraph of the support graph induced by the edges in $E(S)$ is disconnected. This defines a partition of S into sets, say $S_1, \ldots, S_t$. In this case, it is easy to show that the sum of the violations of the balancing inequalities for the sets $S_1, \ldots, S_t$ is equal to the violation of the original balancing inequality. Figure 5 illustrates this possibility. The separation algorithm returns the set $S = \{1, 2, 3, 4\}$, for which the balancing inequality is violated by 4. There are two components, $S_1 = \{1, 2\}$ and $S_2 = \{3, 4\}$, for each of which the balancing inequality is violated by 2.

Thus, if the exact separation algorithm returns a violated balancing inequality, we can often find a number of other violated balancing inequalities with negligible additional work. (A similar observation was made by Nobert and Picard, 1996).

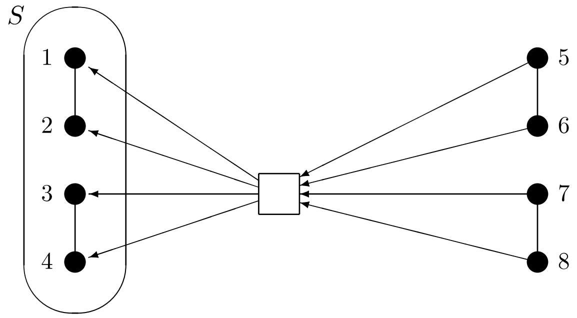

Figure 5 Decomposing a balancing inequality.

The above separation algorithm, using for example the *preflow push* max-flow algorithm of Goldberg and Tarjan (1988), runs in $\mathcal{O}(nm\log(n^2/m))$ time, where m is the number of variables which are positive at (x^*, y^*). Faster heuristics for separation can be devised, based for example on connected components, but this proved to be unnecessary in our computational experiments.

Separation of mixed strengthened comb inequalities

As mentioned above, we described in Lysgaard *et al* (2004) an effective heuristic for the separation of strengthened comb inequalities. This heuristic relied in part on the fact that the separation problem becomes polynomially solvable in the special case in which each tooth has cardinality 2. In this case, the strengthened comb inequalities reduce to a variant of the 2-matching inequalities of Edmonds (1965), and the separation problem can therefore be solved with a minor modification of the algorithm of Padberg and Rao (1982). The separation heuristic for the case of general strengthened comb inequalities can be summarized as follows:

- Apply *shrinking* to the support graph, in such a way that a violated strengthened comb inequality in the shrunk support graph corresponds to a violated strengthened comb inequality in the original.
- Run a heuristic, based on connected components, to generate a set of 'seed' strengthened comb inequalities, not necessarily violated, such that each tooth has cardinality two.
- For each such 'seed' strengthened comb inequality, enlarge the teeth in a greedy way in an attempt to increase the violation (or, equivalently, decrease the slack).
- If no violated inequality has been found, run the modified Padberg–Rao algorithm to find more 'seed' strengthened comb inequalities, and repeat the above greedy procedure.
- If any of the resulting strengthened comb inequalities are violated, output the corresponding violated strengthened comb inequalities for the original graph.

To enable this heuristic to separate the more general mixed strengthened comb inequalities, it was necessary to further modify the Padberg–Rao algorithm, to cope with sending and receiving teeth of cardinality 2. To do this, we note that, for any $i \in V_c$, we have three cases:

- *Case 1*: $q_i < q_{\min}$. In this case, it could be worthwhile making $\{0, i\}$ a tooth in $\mathscr{D}$, and, if we did, it would contribute $1 - (y^*_{0i} + y^*_{i0})$ to the slack of the inequality.
- *Case 2*: $q_i \geqslant q_{\min}$ and $y^*_{i0} \geqslant y^*_{0i}$. In this case, it could be worthwhile making $\{0, i\}$ a tooth in $\mathscr{S}$, which would contribute $1 - y^*_{i0}$ to the slack.
- *Case 3*: $q_i \geqslant q_{\min}$ and $y^*_{i0} < y^*_{0i}$. In this case, it could be worthwhile making $\{0, i\}$ a tooth in $\mathscr{R}$, which would contribute $1 - y^*_{0i}$ to the slack.

On the other hand, if we did not use $\{0, i\}$ as a tooth, but i belonged to the handle, this would contribute $y^*_{0i} + y^*_{i0}$ to the slack. Thus, we take these four possibilities into account when splitting edges in the Padberg–Rao procedure.

In addition, we modified the greedy procedure for enlarging teeth. If a 'seed' tooth contains the depot, we have three options: it can be put into $\mathscr{D}$, $\mathscr{S}$ or $\mathscr{R}$. We run the greedy enlargement process three times accordingly. The chosen category for the tooth is then the one resulting in the minimum slack (ie, the maximum violation).

Separation strategy

In our algorithm for the CVRP, we called our separation routines in the following order. First, we called rounded capacity separation. If this failed to find a rounded capacity violated by a significant amount, we called framed capacity separation. If this also failed, we proceeded to the separation of multistar, strengthened comb and hypotour inequalities. Finally, we also permitted a round of Gomory mixed-integer cuts to be generated.

For the COVRP, we use essentially the same separation strategy. The only differences are that (i) we call both rounded capacity and balancing separation at the first stage, (ii) we replace our strengthened comb separation heuristic with its mixed counterpart, and (iii) to reduce implementational effort, we do not generate Gomory cuts.

Computational experiments

Our algorithm has been coded in the C programming language using the Microsoft Visual C++ v. 6.0 compiler. For solving LPs, we have used the CPLEX callable library v. 9.0. All experiments have been done on a PC with a 1.6 GHz Intel Pentium M processor and 512 MB of RAM running Microsoft Windows XP.

In this section, we show the performance of the entire algorithm on both closed and open versions of several instances. Instead of using our algorithm from Lysgaard *et al* (2004) when solving CVRPs, we use to as large an extent as possible the same code for solving both COVRPs and CVRPs, in order to eliminate the potential effects of differences in implementation details. Specifically, given our COVRP algorithm,

Table 1 Results for the A instances

Name	*COVRP*					*CVRP*	
	Root node		*Branch and cut*			*Branch and cut*	
	LB	*Time*	*LB*	*Tree size*	*Time*	*LB*	*Time*
A-n32-k5	487.306	0.1	487.306*	1	0.1	787.082*	3
A-n33-k5	424.543	0.2	424.543*	1	0.2	662.11*	2
A-n33-k6	455.794	0.4	462.433*	10	0.8	742.693*	1
A-n34-k5	494.313	0.3	508.173*	56	3	780.936*	2
A-n36-k5	510.011	1	519.455*	36	3	802.132*	4
A-n37-k5	482.858	0.5	486.243*	14	0.9	672.465*	3
A-n37-k6	576.64	1	581.073*	14	2	950.852*	52
A-n38-k5	487.597	0.4	497.997*	48	3	733.946*	5
A-n39-k5	541.523	1	549.684*	86	9	828.989*	36
A-n39-k6	526.528	0.3	533.066*	17	2	833.205*	6
A-n44-k7	602.733	1	617.385*	256	29	938.181*	45
A-n45-k6	603.162	2	648.669*	5101	1121	944.876*	13
A-n45-k7	655.541	3	685.156*	5742	1846	1146.772*	1131
A-n46-k7	578.962	0.4	583.538*	6	0.8	917.724*	12
A-n48-k7	651.143	1	669.826*	175	34	1074.338*	46
A-n53-k7	643.795	2	655.184*	70	18	1012.249*	90
A-n54-k7	681.666	4	709.272*	1966	762	1171.682*	2007
A-n55-k9	646.642	2	669.06*	1978	472	1074.464*	43
A-n60-k9	756.362	6	780.409	8048	3600	1341.506	3600
A-n61-k9	639.362	4	664.538	7440	3600	1038.135	3600
A-n62-k8	757.755	4	783.176*	2523	1231	1286.145	3600
A-n63-k9	911.835	10	935.748	5466	3600	1610.838	3600
A-n63-k10	745.936	5	773.995	7692	3600	1303.6	3600
A-n64-k9	811.983	4	837.092	6772	3600	1380.109	3600
A-n65-k9	711.219	3	728.591*	170	58	1181.687*	284
A-n69-k9	732.273	5	757.764*	4644	2659	1152.411	3600
A-n80-k10	1019.84	9	1038.504	4372	3600	1736.875	3600

we obtain a CVRP algorithm by (i) changing c_{i0} from 0 to c_{0i} for $i = 1, \ldots, n$, (ii) deactivating balancing inequalities, and (iii) not permitting sending or receiving teeth in comb inequalities.

We have done our experiments on the so-called `A`, `B`, `E`, `F`, `M`, and `P` benchmark CVRP instances, which are available at www.branchandcut.org (accessed 1 February 2006). Most of the closed versions have already been solved to optimality, see for example Lysgaard *et al* (2004) and Fukasawa *et al* (2006). In all of these instances, the vertices are taken to be points located in the Euclidean plane. The cost of an edge is then taken to be equal to the Euclidean distance between its end-vertices. We had to make a decision concerning *precision* in the computation of these distances. As discussed in Toth and Vigo (2002), some authors represent the distances as floating point numbers within their algorithms, and then report the cost of the solutions to a fixed precision, such as one decimal place. Other authors, however, round the distances to integers following the TSPLIB standard of Reinelt (1991). Our algorithm can cope with both versions. Since papers on the COVRP tend to report results only for the floating point versions, in this paper we do the same.

Tables 1–4 report the results. In each table, the first column gives the instance name. The name indicates the source of the instance, the number of vertices, and the number of vehicles, which for all of these instances is fixed at the minimum possible. For example, `E-n22-k4` is taken from Christofides and Eilon (1969), and has 22 vertices and four vehicles. The next two columns give, for the open version, the lower bound at the root node of the branch-and-bound tree and the time taken to obtain it. All times in the tables are given in seconds. The next three columns give, again for the open version, the best lower bound found within 1 h, the number of branches processed within 1 h, and the time taken to solve the instance to optimality if the instance was solved in less than an hour. An asterisk indicates that the instance has been solved to optimality. The last two columns give the lower bound and time for the closed version of the instance.

In general, it appears that the open version is usually a little easier to solve than the closed version. A remarkable instance is `E-n76-k8`, where the open version can be solved in about 10 min, whereas the closed version is known to be difficult. However, there are some exceptions, such as `F-n135-k7`.

Finally, in Table 5, we use our lower bounds to assess the quality of heuristic methods for the COVRP. For each of 10 COVRP instances, the table reports the lower bound we obtained within 1 h, followed by the upper bounds obtained by five heuristics, those of Brandão (2004), Fu *et al* (2005, 2006),

Table 2 Results for the B instances

Name	*COVRP*					*CVRP*	
	Root node		*Branch and cut*			*Branch and cut*	
	LB	*Time*	*LB*	*Tree size*	*Time*	*LB*	*Time*
B-n31-k5	360.287	0.3	362.725*	6	0.4	676.088*	0.9
B-n34-k5	452.184	6	458.764*	131	13	789.841*	7
B-n35-k5	557.326	0.1	557.326*	1	0.1	956.294*	0.04
B-n38-k6	445.583	0.1	445.628*	2	0.1	807.879*	2
B-n39-k5	317.76	0.6	322.539*	25	1	553.156*	0.2
B-n41-k6	480.432	0.4	483.068*	20	2	833.664*	5
B-n43-k6	428.167	0.4	428.167*	1	0.4	746.694*	7
B-n44-k7	499.447	2	501.308*	26	4	914.965*	0.4
B-n45-k5	484.19	5	488.065*	27	7	753.961*	4
B-n45-k6	390.288	1	403.812*	7645	1536	680.438*	21
B-n50-k7	437.151	0.2	437.151*	1	0.2	744.228*	0.6
B-n50-k8	703.571	4	720.427	12685	3600	1313.548	3600
B-n51-k7	620.415	1	625.14*	75	8	1035.04*	8
B-n52-k7	440.186	0.3	441.193*	12	0.8	749.97*	2
B-n56-k7	415.385	2	420.485*	208	17	712.916*	7
B-n57-k7	639.227	4	646.364*	979	228	1157.731*	152
B-n57-k9	854.691	4	868.025	12972	3600	1602.289*	585
B-n63-k10	826.616	3	837.072*	124	25	1499.096*	276
B-n64-k9	508.17	4	520.466*	1786	499	868.194*	14
B-n66-k9	747.287	4	755.274*	700	248	1320.771	3600
B-n67-k10	604.225	4	616.077	12939	3600	1039.268*	375
B-n68-k9	694.163	6	701.717*	1335	512	1273.673	3600
B-n78-k10	704.604	8	715.399	5778	3600	1225.716	3600

Table 3 Results for the E, F and M instances

Name	*COVRP*					*CVRP*	
	Root node		*Branch and cut*			*Branch and cut*	
	LB	*Time*	*LB*	*Tree size*	*Time*	*LB*	*Time*
E-n22-k4	252.614	0.08	252.614*	1	0.08	375.28*	0.2
E-n23-k3	428.66	0.1	442.984*	22	0.3	568.563*	0.05
E-n30-k3	385.971	0.4	393.512*	56	1	535.797*	2
E-n33-k4	506.346	0.3	511.263*	8	0.6	837.672*	2
E-n51-k5	411.481	1	416.063*	102	16	524.611*	11
E-n76-k7	522.271	10	530.022*	2660	1103	682.563	3600
E-n76-k8	529.371	12	537.239*	1282	636	733.686	3600
E-n76-k10	547.826	21	559.233	4118	3600	818.655	3600
E-n76-k14	602.007	24	614.442	3178	3600	989.257	3600
E-n101-k8	633.851	6	639.744*	3378	2379	820.552	3600
E-n101-k14	692.149	33	699.985	1606	3600	1049.534	3600
F-n45-k4	463.896	0.1	463.896*	1	0.1	723.541*	1
F-n72-k4	175.924	9	176.999*	26	11	241.974*	4
F-n135-k7	753.431	27	762.894	1918	3600	1162.957*	1587
M-n101-k10	528.237	6	534.239*	297	89	819.558*	5
M-n121-k7	647.59	98	657.056	1046	3600	1034.782	3600
M-n151-k12	725.089	70	730.204	1354	3600	992.834	3600

Li *et al* (2006), Pisinger and Ropke (2006) and Tarantilis *et al* (2005). Missing entries indicate either that the authors do not report results on the given instance, or that they report results for a different number of vehicles. It will be seen that the heuristics do rather well and actually produce optimal solutions in a few cases.

Optimal solutions for all COVRP instances solved to optimality are available on the website www.hha.dk/~lys/.

Conclusion

Although the COVRP appears to be a trivial variant of the standard CVRP, we have seen that it is intermediate in

Table 4 Results for the P instances

Name	*COVRP*					*CVRP*	
	Root node		*Branch and cut*			*Branch and cut*	
	LB	*Time*	*LB*	*Tree size*	*Time*	*LB*	*Time*
P-n16-k8	235.06	0.09	235.06*	1	0.09	451.335*	0.4
P-n19-k2	162.586	0.09	168.57*	12	0.2	212.657*	0.1
P-n20-k2	167.814	0.05	170.278*	12	0.2	217.416*	0.1
P-n21-k2	163.877	0.02	163.877*	1	0.02	212.712*	0.03
P-n22-k2	167.191	0.03	167.191*	1	0.03	217.852*	0.3
P-n22-k8	335.27	2	345.867*	22	3	600.826*	2
P-n23-k8	282.472	0.9	302.87*	428	36	531.174*	6
P-n40-k5	347.348	0.4	349.552*	19	2	461.726*	2
P-n45-k5	388.999	2	391.809*	36	4	512.791*	7
P-n50-k7	391.535	1	397.376*	222	24	559.863*	136
P-n50-k8	405.852	2	422.891	10014	3600	628.081	3600
P-n50-k10	426.604	2	440.438*	4704	1072	696.35	3600
P-n51-k10	455.056	3	472.858	9400	3600	741.499*	2225
P-n55-k7	408.572	0.7	411.581*	60	7	570.27*	1599
P-n55-k8	410.584	2	412.554*	13	4	578.61*	207
P-n55-k10	433.56	2	444.308*	603	128	687.141	3600
P-n55-k15	522.954	8	544.341	5358	3600	932.103	3600
P-n60-k10	469.757	6	482.085*	2712	914	741.242	3600
P-n60-k15	542.407	11	560.345	5902	3600	952.636	3600
P-n65-k10	512.073	7	522.501*	2434	1059	789.344	3600
P-n70-k10	536.04	9	548.341	4670	3600	814.949	3600
P-n76-k4	516.857	4	522.945*	804	319	598.196*	37
P-n76-k5	516.57	10	525.635*	2721	1367	633.317*	3124
P-n101-k4	619.495	19	621.749*	56	45	691.287*	47

Table 5 Lower bounds compared with upper bounds from the literature

Name	*Our LB*	*Brandao*	*Fu et al*	*Li et al*	*P & R*	*T et al*
E-n51-k5	416.1*	416.1*	416.1*	416.1*	416.1*	—
E-n76-k10	559.2	574.5	567.1	567.1	567.1	—
E-n101-k8	639.7*	641.6	641.9	639.7*	641.8	—
M-n101-k10	534.2*	535.1	534.7	534.2*	534.2*	534.2*
M-n121-k7	657.1	683.4	717.2	682.5	682.1	—
M-n151-k12	730.2	740.8	738.9	733.1	733.1	733.7
M-n200-k16	848.5	953.4	—	925.0	896.1	—
M-n200-k17	847.6	—	879.0	—	—	870.3
F-n72-k4	177.0*	177.4	177.0*	177.0*	177.0*	—
F-n135-k7	762.9	781.2	777.1	769.7	770.2	—

generality between the CVRP and the ACVRP. As a result, some subtle modifications are needed to adapt a branch-and-cut code for the CVRP to the COVRP. This includes modifications to the formulation, additional classes of inequalities, and adjustments to the separation algorithms.

Our results show that small- to medium-scale instances of the COVRP are just as amenable to exact solution by branch-and-cut as their CVRP counterparts. In fact, if anything, the open versions often appear to be slightly easier.

Future research could include the incorporation of column generation, leading to a full branch-cut-and-price algorithm along the lines of the one presented in Fukasawa *et al* (2006). This would no doubt lead to the solution of even more instances, especially those with small vehicle capacities and a large number of vehicles.

References

Aksen D, Özyurt Z and Aras N (2006). The open vehicle routing problem with driver nodes and time deadlines. *J Opl Res Soc*, AOP 9/8/06 doi: 10.1057/palgrave.jors.2602249.

Augerat P (1995). *Approche Polyèdrale du Problème de Tournées de Véhicules*. PhD thesis, Institut National Polytechnique de Grenoble.

Bodin LD, Golden BL, Assad AA and Ball MO (1983). Routing and scheduling of vehicles and crews: The state of the art. *Comput Opns Res* **10**: 63–211.

Brandão J (2004). A tabu search algorithm for the open vehicle routing problem. *Eur J Opl Res* **157**: 552–564.
Christofides N and Eilon S (1969). An algorithm for the vehicle-dispatching problem. *Opns Res Quart* **20**: 309–318.
Edmonds J (1965). Maximum matching and a polyhedron with 0–1 vertices. *J Res Nat Bur Standards* **69B**: 125–130.
Fischetti M, Toth P and Vigo D (1994). A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Opns Res* **42**: 846–859.
Ford L and Fulkerson D (1962). *Flows in Networks*. Princeton University Press: New Jersey.
Fu Z, Eglese R and Li LYO (2005). A new tabu search heuristic for the open vehicle routing problem. *J Opl Res Soc* **56**: 267–274.
Fu Z, Eglese R and Li LYO (2006). Corrigendum to 'A new tabu search heuristic for the open vehicle routing problem'. *J Opl Res Soc* **57**: 1018.
Fukasawa R *et al* (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Math Program* **106**: 491–511.
Goldberg AV and Tarjan RE (1988). A new approach to the maximum flow problem. *J ACM* **35**: 921–940.
Laporte G, Nobert Y and Desrochers M (1985). Optimal routing under capacity and distance restrictions. *Opns Res* **33**: 1050–1073.
Letchford AN, Eglese RW and Lysgaard J (2002). Multistars, partial multistars and the capacitated vehicle routing problem. *Math Program* **94**: 21–40.
Li F, Golden B and Wasil E (2006). The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Comput Opns Res*, doi: 10.1016/j.cor.2005.11.018.
Lysgaard J (2003). *CVRPSEP*: *A Package of Separation Routines for the Capacitated Vehicle Routing Problem*. Working Paper 03-04, Department of Management Science and Logistics, Aarhus School of Business. www.asb.dk/~lys/.
Lysgaard J, Letchford AN and Eglese RW (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math Program* **100**: 423–445.
Naddef D and Rinaldi G (2002). Branch-and-cut algorithms for the capacitated VRP. In: Toth and Vigo (eds). *The Vehicle Routing Problem. Chapter 3*. SIAM Monographs on Discrete Mathematics and Applications: Philadelphia PA.
Nobert Y and Picard J-C (1996). An optimal algorithm for the mixed Chinese postman problem. *Networks* **27**: 95–108.
Padberg MW and Rao MR (1982). Odd minimum cut-sets and *b*-matchings. *Math Opns Res* **7**: 67–80.
Pisinger D and Ropke S (2006). A general heuristic for vehicle routing problems. *Comput Opns Res*, doi: 10.1016/j.cor.2005.09.012.
Reinelt G (1991). TSPLIB—A traveling salesman problem library. *ORSA J Comput* **3**: 376–384.
Repoussis PP, Tarantilis CD and Ioannou G (2006). The open vehicle routing problem with time windows. *J Opl Res Soc*, AOP 1/2/06 doi: 10.1057/palgrave.jors.2602143.
Sariklis D and Powell S (2000). A heuristic method for the open vehicle routing problem. *J Opl Res Soc* **51**: 564–573.
Schrage L (1981). Formulation and structure of more complex/realistic routing and scheduling problems. *Networks* **11**: 229–232.
Tarantilis CD, Diakoulaki D and Kiranoudis CT (2004). Combination of geographical information system and efficient routing algorithms for real life distribution operations. *Eur J Opl Res* **152**: 437–453.
Tarantilis CD, Ioannou G, Kiranoudis CT and Prastacos GP (2005). Solving the open vehicle routeing problem via a single parameter metaheuristic algorithm. *J Opl Res Soc* **56**: 588–596.
Toth P and Vigo D (eds). *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications: Philadelphia PA.

Appendix A

To show validity of the mixed comb inequalities, it is helpful to prove the following lemma.

Lemma 1 *For any set S such that* $0 \in S$, *the following three inequalities are valid.*

$$x(E(S\backslash\{0\})) + y^+(S\backslash\{0\}) + y^-(S\backslash\{0\}) \leqslant |S| + K - k(V\backslash S) - 1 \quad (A.1)$$

$$x(E(S\backslash\{0\})) + y^+(S\backslash\{0\}) \leqslant |S| - 1 \quad (A.2)$$

$$x(E(S\backslash\{0\})) + y^-(S\backslash\{0\}) \leqslant |S| - 1 \quad (A.3)$$

Proof Owing to the degree equations, the inequality (A.1) is equivalent to the capacity inequality on $V\backslash S$ and the inequalities (A.2) and (A.3) are equivalent to the balancing inequalities on $V\backslash S$ and $S\backslash\{0\}$, respectively. □

Proof of Theorem 1 We follow the standard Chvátal–Gomory integer rounding argument. If we sum together the following inequalities:

the degree equations for all $i \in H$,
the inequality (6) on $H \cap T_i$ for $1 \leqslant i \leqslant t$,
the inequality (6) on T_i and $T_i\backslash H$ for $i \in \mathcal{N}$,
the inequalities (A.1) for T_i and $T_i\backslash H$, for $i \in \mathcal{D}$,
the inequalities (A.2) for T_i and $T_i\backslash H$, for $i \in \mathcal{S}$, and
the inequalities (A.3) for T_i and $T_i\backslash H$, for $i \in \mathcal{R}$,

we obtain (after some re-arranging):

$$2x(E(H)) + 2\sum_{i\in\mathcal{N}} x(E(T_i)) + 2\sum_{i\in\mathcal{D}\cup\mathcal{S}\cup\mathcal{R}} x(E(T_i\backslash\{0\})) + 2\sum_{i\in\mathcal{D}\cup\mathcal{S}} y^+(T_i\backslash\{0\}) + 2\sum_{i\in\mathcal{D}\cup\mathcal{R}} y^-(T_i\backslash\{0\}) + x\left(\delta(H)\Big\backslash\bigcup_{i=1}^{t} E(T_i)\right) + y^+\left(H\Big\backslash\bigcup_{i=1}^{t} T_i\right) + y^-\left(H\Big\backslash\bigcup_{i=1}^{t} T_i\right) \leqslant 2|H| + 2\sum_{i=1}^{t}|T_i| + 2|\mathcal{D}|(K-1) - \sum_{i=1}^{t}\gamma(T_i).$$

Dividing this inequality by two and rounding down yields the result. □

Received February 2006;
accepted October 2006 after one revision