Application of the Branch and Cut Method to the Vehicle Routing Problem

Ulrich Blasum
Universität zu Köln
blasum@zpr.uni-koeln.de

Winfried Hochstättler
BTU Cottbus
hochst@math.tu-cottbus.de

May 15, 2002

Abstract

The successful application of Branch and Cut methods to the TSP has drawn attention also to the polyhedral properties of the symmetric capacitated vehicle routing problem, which is the capacitated counterpart of the TSP. We investigate three classes of valid inequalities for the CVRP, multistars, pathbin inequalities and hypotours and give computational results we obtained with a Branch and Cut implementation.

1 Introduction

The symmetric capacitated vehicle routing problem (CVRP) with homogeneous vehicle fleet combines ingredients of the Traveling Salesman Problem (TSP) and the Bin-Packing Problem (BPP). While the TSP is well suited for the polyhedral approach of the Branch and Cut algorithm [JRR95], the BPP has been treated much more successfully by intelligent direct enumeration of the solution space [MT90]. This difference in behavior is mirrored by the numerical results currently obtained when applying Branch and Cut to the CVRP. While the method is basically suited to obtain exact solutions for the problem, the instances that can be solved are magnitudes smaller than those of the TSP. Trivial TSP instances turn out to be a real challenge as soon as a capacity restriction is introduced.

Nevertheless the CVRP is practically repulsive to all exact methods, like Lagrangian relaxation [FJ81] or set partitioning approaches [YKH89, HCM95], which have been applied to it. So far, the largest instances have been solved with Branch and Cut [ABB+95], so it seems worth following this path. This requires a thorough knowledge of the polyhedral properties of the problem and also improvements of the separation heuristics for the inequalities known so far.

A general survey on the vehicle routing problem is given in the book [BMMN95]. Results on facets of the CVRP have first been published by Araque [AHM90, Ara90] for the unit demand CVRP and by Cornuéjols and Harche [CH93] for the general demand case. A survey on Branch and Cut for the CVRP can be found in [NR01]. Many valuable contributions on valid inequalities for the CVRP have been made by Augerat (see [NR01]). A survey on linear relaxations of the CVRP is given in [Lap92].

In this paper we make both theoretical and computational contributions to the Branch and Cut approach by inspecting three classes of valid inequalities. In the first part we develop an exact identification routine for violated multistar inequalities in the undirected CVRP and derive a new compact and strong linear relaxation for the CVRP from this separator. We will also introduce a generalization of the multistar inequalities. In Section 4 we review the pathbin inequalities presented by Augerat. We introduce a special case of these inequalities which is much easier to handle in practice than the general form. We will also prove a sufficient condition for the pathbin inequalities to define facets of the Graphical Vehicle Routing Problem (GVRP).

In Section 5 we propose several improvements of a heuristic suggested in [ABB+95] to identify violated hypotour inequalities. Furthermore, we suggest a method to combine hypotour inequalities to new valid inequalities. We give numerical results obtained with our separation routine and close the paper by presenting computational results of our Branch and Cut implementation. We were able to solve two of the difficult 76-node benchmarks from Eilon [CE69] to optimality.

2 Basic Definitions

An instance of the symmetric *Capacitated Vehicle Routing Problem (CVRP)* with homogeneous vehicle fleet is given by

1. a set of customers V_0 with demands $d_i \in \mathbb{N}$ $(i \in 1, \dots, |V_0|)$, and a depot v_0, $V := V_0 \cup \{v_0\}$,
2. a vehicle capacity $C \in \mathbb{N}_+$, a number $k \in \mathbb{N}_+$, denoting the number of routes,
3. and a symmetric distance function $c_{ij} \in \mathbb{N}_+$ on the edge set $E \subseteq V \times V$.

We assume the underlying undirected graph $G := (V, E)$ to be complete. A *tour* is either a simple cycle in G containing v_0 or a closed walk of the form (v_0, v, v_0) if there is only one customer $v \in V_0$ served by the vehicle.

The objective is to find a set of k nonempty tours, of minimal total length, such that each tour contains the depot, each customer is served by exactly one tour and the total demand of the customers in each tour does not exceed the vehicle capacity C.

A solution of the CVRP can be represented by a family of edges and is called a simple k–tour. We will relax notation by dropping the word simple in the following. Non-simple tours will be considered only in Section 4.

As usual, if M is a set, $x : M \to \mathbb{R}$ is a function, and $S \subseteq M$, we use the notation $x(S) := \sum_{v \in S} x_v$. If $S \subseteq V$ we, furthermore, use $\gamma(S) := \{(u, v) \in E : u, v \in S\}$ and denote its coboundary by $\delta(S) := \{e = (i, j) \in E : i \in S \not\ni j\}$. For $S, T \subseteq V$ we denote $(S : T) := \{e = (i, j) \in E : i \in S, j \in T\}$. For notational reasons we define $d_0 := 0$.

We use $r(S)$ to denote the minimum number of bins of capacity C needed to pack all demands in S, which is a natural lower bound for the number of tours needed to serve the customers in $S \subseteq V_0$.

The symmetric CVRP can then be modeled by the following integer program.

$$
\begin{aligned}
& & \min(c^T x) & & \quad (1)\\
& & x(\delta(v)) = 2 & \qquad \forall v \in V_0 & \quad (2)\\
& & x(\delta(v_0)) = 2k & & \quad (3)\\
\text{s.t.} & & x(\delta(S)) \geq 2r(S) & \qquad \forall S \subseteq V_0 & \quad (4)\\
& & x_e \in [0, 1] & \qquad \forall e \in V_0 \times V_0 & \quad (5)\\
& & x_e \in [0, 2] & \qquad \forall e \in \delta(v_0) & \quad (6)\\
& & x & \qquad \text{integer} & \quad (7)
\end{aligned}
$$

Clearly, the feasible solutions x of this program coincide with the incidence vectors of the sets of edges of k–tours. Equalities (2) and (3) are the degree-constraints for each customer and for the depot. Inequalities (4) are called capacity constraints. They are a tightening of the *subtour elimination constraints* "$x(\delta(S)) \geq 2$" known from the TSP to the CVRP guaranteeing that the vehicle capacity is not exceeded in any tour.

The GVRP is the relaxation of the CVRP which is obtained when the tours are not required to be simple cycles but can be arbitrary closed walks, which may use edges multiply and which can also revisit customers or visit customers which are not assigned to that tour. Still each customer must be assigned to exactly one tour and the capacity restriction only affects the customers which are actually served by the tour. It is known that the $CVRP$-polyhedron is a face of the $GVRP$-polyhedron and that the optimal solutions of a GVRP instance are from this face as long as the underlying graph is complete and the triangle inequality holds ([CFN85]). The GVRP is the capacitated version of the GTSP. The investigation of the GTSP (e.g. [CFN85]) has helped to gain insight into the facial structure of the TSP, since it simplifies the analysis considerably. In a similar way the GVRP is used for the analysis of the CVRP.

3 Multistar Inequalities

3.1 Formulation

The multistar constraints were originally introduced in [Gav84] for the directed capacitated minimal spanning tree problem and then transferred to the directed [Gou95] and to the undirected CVRP [LE96]. In this section we will review and generalize them. The corresponding separation problem is modeled as a network flow problem yielding a polynomial time separation routine. A careful analysis of the network and its capacities leads to a new linear relaxation for the CVRP.

The multistar inequalities can be formulated as follows:

$$x(\delta(S)) \geq 2\frac{d(S)}{C} + 2\sum_{\substack{i \notin S \\ j \in S}} \frac{d_i}{C} x_{ij} \qquad \forall S \subseteq V_0. \tag{8}$$

The rationale behind them is that a vehicle visiting the customers in S and using an edge (i,j) with $i \notin S, j \in S$ must have sufficient free capacity for $S \cup \{v_i\}$. The same consideration can be applied if more than only one vehicle visits customers in S. We denote by $CVRP_{MS}$ the polytope obtained from (2), (3), (5), (6) and (8).

Replacing the vertices in the neighborhood of $S = S_0$ by sets of vertices we derive a generalization of the multistar inequalities as follows:

Lemma 3.1. *Let $S_0, \ldots, S_t$ be pairwise disjoint nonempty subsets of V_0 with $d(S_i) \leq C$ for $i = 1, \ldots, t$. Then the* setbased multistar inequality

$$x(\delta(S_0)) \geq 2\frac{d(S_0)}{C} + 2\sum_{i=1}^{t} \frac{d(S_i)}{C} \left(x\left(S_0 : S_i\right) + 2 - x\left(\delta\left(S_i\right)\right)\right) \tag{9}$$

is valid for the CVRP.

Proof. Let x be a feasible solution for the CVRP, i.e. x satisfies (2) to (7) and $S_1, \ldots, S_t$ be as in the assumptions of the theorem. Assume w.l.o.g. that $S_1, \ldots, S_s$ are those sets out of $S_1, \ldots, S_t$ for which

$2x(S_0 : S_i) - x(\delta(S_i)) \geq 0$. Then we obtain (9) from the following inequalities, explained below:

$$
\begin{aligned}
x(\delta(S_0)) &= x(S_0 : E \setminus \bigcup_{i=0}^{s} S_i) + \sum_{i=1}^{s} x(S_0 : S_i) && (10)\\
&= x(\delta(\bigcup_{i=0}^{s} S_i)) - \sum_{i=1}^{s} x(\delta(S_i)) + \sum_{i=1}^{s} \left(x(S_0 : S_i) + x(S_i : \bigcup_{\substack{j=1\\ j\neq i}}^{s} S_j) \right) + \sum_{i=1}^{s} x(S_0 : S_i) && (11)\\
&= x(\delta(\bigcup_{i=0}^{s} S_i)) + \sum_{i=1}^{s} \left(2x(S_0 : S_i) - x(\delta(S_i)) + x(S_i : \bigcup_{\substack{j=1\\ j\neq i}}^{s} S_j) \right) && (12)\\
&\geq x(\delta(\bigcup_{i=0}^{s} S_i)) + \sum_{i=1}^{s} (2x(S_0 : S_i) - x(\delta(S_i))) && (13)\\
&\geq 2\sum_{i=0}^{s} \frac{d(S_i)}{C} + 2\,|\{i : i = 1, \dots, s; x(S_0 : S_i) = x(\delta(S_i)\}| && (14)\\
&\geq 2\sum_{i=0}^{s} \frac{d(S_i)}{C} + 2\sum_{i=1}^{s} \frac{d(S_i)}{C}(1 + x(S_0 : S_i) - x(\delta(S_i))) && (15)\\
&\geq 2\frac{d(S_0)}{C} + 2\sum_{i=1}^{t} \frac{d(S_i)}{C}(2 + x(S_0 : S_i) - x(\delta(S_i))). && (16)
\end{aligned}
$$

The first equalities are seen by counting edges, (14) makes use of the fact that any cut in the Eulerian graph induced by a k–tour has even cardinality, whereas (15) exploits integrality only. Finally we get (16) from

$$
\begin{aligned}
& & 2x(S_0 : S_i) - x(\delta(S_i)) &< 0\\
&\Leftrightarrow & 2x(S_0 : S_i) - x(\delta(S_i)) &\leq -2\\
&\Rightarrow & x(S_0 : S_i) + 2 - x(\delta(S_i)) &\leq 0.
\end{aligned}
$$

□

The setbased multistar inequalities are useful in practice in the following special case.

Corollary 3.2. *Let x be a vector satisfying* (2), (3), (5) *and* (6) *that violates a multistar constraint but does not violate a subtour elimination constraint. For $i = 1, \dots, t$ let $S_i \subseteq V_0$ denote the maximal node sets satisfying $x(\delta(S_i)) = 2$. Let $G/\mathcal{S}$ denote the graph obtained by shrinking each of $S_i, i = 1, \dots, t$ to a single vertex adjusting the demand function to $d(S_i)$. Then the vector $x_{|G/\mathcal{S}}$ violates a multistar constraint in $G/\mathcal{S}$. This violated inequality in the shrunk support graph corresponds to a violated setbased multistar constraint in the original graph.*

Proof. Let x be such a vector that violates a multistar constraint with respect to the set S. Choose such an S such that $|S|$ is as large as possible. Since x does not violate a subtour elimination constraint all $\delta(S_i)$ are mincuts. Therefore, as unions and intersections of mincuts are mincuts again (see e.g. [AMO93]), the maximal node sets $S_i \subseteq V_0$ satisfying $x(\delta(S_i)) = 2$ must be pairwise disjoint and, thus, partition V_0. Furthermore, by the choice of S, all S_i are either disjoint from or a subset of S. Let S_0 be the set of (super) nodes in $G/\mathcal{S}$ corresponding to vertices from S. Then

$$\begin{aligned} x_{|G/\mathcal{S}}(\delta(S_0)) &= x(\delta(S)) \\ &< 2\frac{d(S)}{C} + 2\sum_{\substack{k\in S\\ j\notin S}} \frac{d_j}{C} x_{kj} \\ &\leq 2\frac{d(S)}{C} + 2\sum_{S_i\cap S=\emptyset} \frac{d(S_i)}{C}\left(x\left(S_0:S_i\right)+2-x\left(\delta\left(S_i\right)\right)\right). \end{aligned}$$

□

3.2 Separation

We will now show that a violated multistar constraint can be identified using a network flow algorithm. This approach is similar to the one already known for capacity constraints of the form

$$x(\delta(S)) \geq 2\frac{d(S)}{C}. \tag{17}$$

We will therefore start by shortly reviewing the corresponding separation routine (see [NR01] for reference).

3.2.1 Separation of Capacity Constraints

The basic idea of the algorithm to identify violated capacity constraints of the type (17) presented by Harche and Rinaldi [ABB+95, NR01] can be described as follows. First consider the case of the directed CVRP. When x is a feasible point then it is possible to simultaneously route, through a network of capacity Cx, d_i units of flow from the depot to each node. The flow on each arc corresponds to the load of the vehicle. In the case of the undirected CVRP we construct a network D by introducing for each undirected edge $(ij) \in E$ two directed arcs of capacity $u_{ij} = u_{ji} = \frac{1}{2}Cx_{ij}$. Again it must be possible to simultaneously route d_i units of flow to each node. In order to model this as a maximum flow problem, we add to D a sink node t and an arc (it) of capacity d_i from every node to the sink.

The value of a maximum flow in this network is $\sum_{i\in V} d_i$ if and only if x satisfies (17) for all $S \subseteq V_0$. Otherwise we find a (v_0,t)-cut $(V\setminus S : S)$ of smaller capacity and the violated inequality $x(\delta(S)) \geq 2\left\lceil \frac{d(S)}{C}\right\rceil$.

3.3 Separation of multistars

Generalizing this idea, the separation problem for the polytope $CVRP_{MS}$ can be reduced to a network flow problem by a modification of the arc capacities in D. We will first restrict our attention to the formally easier case where all demands satisfy $d_i \leq \frac{1}{2}C$. In this case we change the capacities to $u_{ij} = u_{ji} = (\frac{1}{2}C - d_i)x_{ij}$ for each arc in the network not connected to t. The capacities for the sink arcs remain unchanged. Now assume we have a (v_0, t)-Cut $V \setminus S : S$ of value strictly smaller than $d(V_0)$. Then

$$\begin{aligned} 0 &> u(V \setminus S : S) - \sum_{i \in V} d_i && (18) \\ &= \sum_{i \in V \setminus S} d_i + \sum_{\substack{i \in S; i \neq t \\ j \notin S}} \left(\frac{1}{2}C - d_j\right) x_{ji} - \sum_{i \in V} d_i && (19) \\ &= \sum_{\substack{i \in S; i \neq t \\ j \notin S}} \left(\frac{1}{2}C - d_j\right) x_{ji} - \sum_{i \in S; i \neq t} d_i. && (20) \end{aligned}$$

Such a cut thus yields a violated multistar inequality for the set $S \setminus \{t\}$ and each violated multistar inequality yields a cut with capacity less than $d(V_0)$. The capacities in this network would be negative when $d_i > \frac{1}{2}C$ for some node i and thus the network must be modified. Informally, we can regard this modification as an initial flow, which distributes the "overflow" $\max\{0, d_i - \frac{1}{2}C\}$ to each node adjacent to i in the network defined by the support graph of the current point x. For $d \in [0, C]$ we define $\underline{d}$ resp. $\overline{d}$ as the fraction of d being smaller resp. larger than $\frac{1}{2}C$:

$$\begin{aligned} \underline{d} &:= \min\{\frac{1}{2}C, d\} \\ \overline{d} &:= d - \underline{d}. \end{aligned}$$

The new capacities in the network are given by

$$u_{ij} = \begin{cases} \left(\frac{1}{2}C - \underline{d}_i - \overline{d}_j\right) x_{ij} & \forall (i,j) \in E; i, j \neq v_0, t \\ \frac{1}{2}C x_{v_0 j} + \max\{0, \sum_{k \in \delta(j); k \neq v_0, t} (\overline{d}_j - \overline{d}_k) x_{jk}\} & \forall i = v_0, j \in V_0 \\ d_i + \max\{0, \sum_{k \in \delta(i); k \neq v_0, t} (-\overline{d}_i + \overline{d}_k) x_{ik}\} & \forall i \in V_0, j = t \\ 0 & \text{if } (i,j) = (v_0, t). \end{cases} \qquad (21)$$

It is crucial to note that no capacities can become negative. It is easy to check that the factors

$$\frac{1}{2}C - \min\{\frac{1}{2}C, d_j\} - (d_i - \min\{\frac{1}{2}C, d_i\})$$

are positive as long as $d_i + d_j \leq C$ holds. When this condition is violated the corresponding variable can be fixed to zero in a preprocessing since the two customers can not be served in the same tour.

Again we compute the value of a v_0, t-cut $V \setminus S : S$ with $t \in S$. For simplicity we define $S' := S \setminus \{t\}$.

$$\begin{aligned} u(V \setminus S : S) =& u(V_0 \setminus S' : S') + u(v_0 : S') + u(V_0 \setminus S' : t) \\ =& u(V_0 \setminus S' : S') + u(v_0 : S') - u(S' : t) + u(S' : t) + u(V_0 \setminus S' : t) \\ =& u(V_0 \setminus S' : S') + u(v_0 : S') - u(S' : t) + u(V_0 : t) \\ =& \sum_{\substack{i \in V_0 \setminus S' \\ j \in S'}} \left(\frac{1}{2}C - \underline{d}_i - \overline{d}_j\right) x_{ij} + \\ & \sum_{j \in S'} \left(\frac{1}{2}C x_{v_0 j} - d_j + \sum_{i \in \delta(j), i \neq v_0, t} (\overline{d}_j - \overline{d}_i) x_{ji}\right) + u(V_0 : t) \\ =& \frac{1}{2}C x(V \setminus S' : S') - d(S') - \sum_{\substack{i \in V_0 \setminus S' \\ j \in S'}} d_i x_{ij} + const. \end{aligned}$$

The constant $u(V_0 : t)$ is independent from the cut under consideration. Again a cut with a value lower than this constant corresponds to a violated multistar inequality, and vice versa any violated inequality induces such a cut.

Thus the multistar inequalities can be separated by computing a minimum v_0, t-cut in this network. We thus have the result

Corollary 3.3. *The multistar inequalities for the $CVRP$-polytope can be separated in polynomial time.*

Remark 3.4. *The preceding mincut-algorithm does not make use of the degree-inequalities. It can thus be modified in order to separate the multistar-inequalities also in the case of the symmetric capacitated minimum spanning tree problem. In this case the capacities from the depot to each node and the sink-arcs have to be modified. Since this requires some formalism we will not go into the details.*

It should be noted that the multistar inequalities dominate the inequalities (17). Nevertheless they are not directly useful in a Branch and Cut framework since they are rarely supporting for the $CVRP$ polytope. In our implementation of a cutting plane algorithm for the CVRP we have therefore used the multistar separator only as part of a heuristic to identify violated capacity constraints of the form

$$x(\delta(S)) \geq 2\left\lceil \frac{d(S)}{C} \right\rceil . \tag{22}$$

Given a violated multistar inequality with set S we first check whether (22) is violated. If this is not the case we enlarge S by a shrinking heuristic as described in [ABB$^+$95] in order to find a violated capacity constraint. This heuristic successively adds the node v which maximizes $x(v : S)$ to the initial set S.

Using Corollary 3.2 the separation routine can be improved by shrinking, in a preprocessing step, all maximal subsets $S_1, \ldots, S_l$ of V with $x(\delta(S_i)) = 2$. This can be done in polynomial time since this amounts to finding successively all minimum cuts in the support graph separating the depot from some node v with maximum sized shore on the v side. In this shrunken network the algorithm finds violated valid inequalities of the form (9) with the S_i being the nodes of the shrunk graph. Clearly there is a violated multistar inequality in the shrunk graph whenever there is one in the original graph.

3.4 LP Formulation

The separation-algorithm described in the preceding section motivates a linear relaxation of the polytope $CVRP$.

The standard linear programming formulation of the network flow problem from the last section, with flow variables y_{ij} has the following form:

$$\begin{array}{lllr}
 & \min(c^T x) & & \\
 & x(\delta(i)) = 2 & \forall_{i \in V_0} & (23)\\
\text{s.t.} & x(\delta(v_0)) = 2k & & (24)\\
 & \sum_{j\in\delta(i)} y_{ji} - y_{ij} = d_i + \max\{0, \sum_{k\in\delta(i);k\neq v_0} (-\overline{d}_i + \overline{d}_k)x_{ik}\}, & \forall_{i \in V_0} & (25)\\
 & y_{v_0j} \leq \frac{1}{2}Cx_{v_0j} + \max\{0, \sum_{k\in\delta(j);k\neq v_0} (\overline{d}_j - \overline{d}_k)x_{jk}\} & \forall (v_0j) \in E & (26)\\
 & y_{ij} \leq \left(\frac{1}{2}C - \underline{d}_i - \overline{d}_j\right) x_{ij} & \forall (i,j) \notin \delta(v_0) & \\
 & 0 \leq y_e & \forall e \in E & \\
 & x_e \in [0,1] & \forall e \notin \delta(v_0) & \\
 & x_e \in [0,2] & \forall e \in \delta(v_0) &
\end{array}$$

Name	$CVRP_{L_1}$	$CVRP_{MS(L_1)}$	LB $CVRP_{cap}$	Best Sol.
E-n101-k8	757.48	768.92	796.424	817
E-n22-k4	335.29	349.97	375	375
E-n23-k3	521.85	545.36	569	569
E-n30-k3	431.2	448.74	508.5	534
E-n51-k5	489.48	499.43	514.524	521
E-n76-k10	735.91	771.77	789.451	835
E-n76-k7	630.08	644.3	661.361	682
E-n76-k8	667.43	690.47	711.213	735
F-n135-k4	979.23	1009.21	1158.25	1165
F-n45-k4	584.54	611.84	724	724
F-n72-k4	203.27	205.05	232.5	238
M-n101-k10	744.55	769.78	> 819	820

Table 1: Lower bound from some LP-relaxations of the $CVRP$.

From the last section it can be readily concluded, that the $x - values$ of the feasible points of that program must satisfy all multistar inequalities. Vice versa, a maximal flow in the network from the last section shows, that all points of $CVRP_{MS}$ can be extended to feasible points for this program. In order to eliminate the maximum operator, and thus to obtain a linear program the flow equalities (25) are solved for $y_{v_0 i}$ (see (34)) and these variables are substituted in (26).

With these operations we obtain the following lift of $CVRP_{MS}$:

$$
\begin{aligned}
x(\delta(i)) &= 2 && \forall_{i \in V_0} && (27)\\
x(\delta(v_0)) &= 2k && && (28)\\
\sum_{j \in \delta(i): j \neq v_0} y_{ji} - y_{ij} &\geq d_i - \frac{1}{2} C x_{v_0,i} + \sum_{k \in \delta(i); k \neq v_0} \left(-\overline{d}_i + \overline{d}_k\right) x_{ik}, && \forall_{i \in V_0} && (29)\\
y_{ij} &\leq \left(\frac{1}{2} C - \underline{d}_i - \overline{d}_j\right) x_{ij} && \forall (i,j) \notin \delta(v_0) && (30)\\
0 &\leq y_e && \forall e \in E && (31)\\
x_e &\in [0,1] && \forall e \notin \delta(v_0) && (32)\\
x_e &\in [0,2] && \forall e \in \delta(v_0). && (33)
\end{aligned}
$$

Note, that in the latter program there may exist feasible points (x, y) such that

$$
y_{v_0,i} := d_i + \max\{0, \sum_{k \in \delta(i); k \neq v_0} (-\overline{d}_i + \overline{d}_k) x_{ik}\} - \sum_{j \in \delta(i): j \neq v_0} y_{ji} - y_{ij} < 0. \tag{34}
$$

By standard network flow techniques it is easy, though, to construct a feasible point $(x, \tilde{y})$ satisfying $\tilde{y}_{v_0,i} \geq 0$.

Table 1 compares the bounds derived from this relaxation, shown in column three, to two other LP-bounds. The second column lists the optimal value of the relaxation of the LP introduced in Section 2 where $r(S)$ has been replaced by $\frac{d(S)}{C}$. In column four the latter was replaced by $\lceil \frac{d(S)}{C} \rceil$. These values were computed by an exact enumeration. (We note that these values may be slightly biased as they have been computed within our cutting plane code which fixes edges to zero due to reduced cost arguments and logical implications.) The instances are taken from [Aug], a compilation extending the vehicle routing instances found in the TSPLIB of Reinelt [Rei91].

4 Pathbin Inequalities

Beyond subtour elimination constraints, comb inequalities are the second class of facet defining inequalities used in successful Branch and Cut codes for the TSP. Laporte and Norbert [LN84] have developed capacitated versions of the comb inequalities by replacing the various subtour elimination constraints, used to derive them, by their capacitated counterpart. This approach has helped to incorporate the depot into the formulation but it did not yield any strong capacitated comb inequalities when the depot is not touched. The class of pathbin inequalities introduced by Augerat ([NR01] and references therein) contains a stronger capacitated version of the comb inqualities as a special case. In this section we investigate this special case in more detail and give a suffient condition for it to define a facet of the $GVRP$.

Here, we give a slightly less general description of the pathbin inequalites than Augerat's original one in order to avoid notational difficulties. The inequalities are defined by a handle $H \subseteq V_0$,teeth $T_1, \ldots, T_s \subseteq V_0$ and spots $T_{s+1}, \ldots, T_{s+t} \subseteq H$. Each tooth must intersect the handle without being a subset of it. All teeth and spots are pairwise disjoint and satisfy $d(T_i) \leq C$ for $i = 1, \ldots s+t$. The number $r'(H|T_1, \ldots, T_{s+t})$ is defined as the solution of a constrained bin packing problem with one item for each tooth or spot T_i of height $d(T_i)$ and one item for each node from $v \in H \setminus \bigcup_{i=1}^{s+t} T_i$ of height $d(\{v\})$. The bin capacity is C. The solution of the bin packing problem must satisfy the additional constraint, that each bin contains at most two items corresponding to a tooth from $T_1, \ldots T_s$.

Then the pathbin inequality

$$x(\delta(H)) + \sum_{i=1}^{s+t} x(\delta(T_i)) \geq 2r'(H|T_1, \ldots, T_{s+t}) + 2(s+t) \tag{35}$$

is valid for the $CVRP$.

We will from now on assume that there are no spots, i.e. $t = 0$. A drawback of the pathbin inequalities results from the difficulty of determining $r'(H|T_1, \ldots, T_{s+t})$. A relaxed version can be obtained from the following construction. Consider a graph F_a with one node for every tooth and an edge between any two nodes which represent teeth T_i, T_j with $d(T_i \cup T_j) \leq C$.

Theorem 4.1. *Let $\mathcal{C} := \{H, T_1, \ldots T_s\}$ be a family of node sets with a handle H and teeth $T_1, \ldots, T_s$ satisfying the above conditions and $s \geq 3$. Let the graph F_a be defined as above and let M be a maximum cardinality matching in F_a. Then the inequalities*

$$x(\delta(H)) + \sum_{i=1}^{s} x(\delta(T_i)) \geq 4s - 2|M| \tag{36}$$

are valid for the CVRP.

Proof. The number $s - |M|$ is a lower bound for $r'(H|T_1, \ldots, T_s)$. Thus the inequalities are dominated by the pathbin inequalities. □

Inequality (36) reduces to the inequality of Lemma 5.1 in [Ara90] in the case of unit weights. Araque introduces the term "large teeth" for those teeth which can not be served together with any other tooth by one tour. The formulation with matchings in the graph F_a is slightly more general and in some cases also stronger.

Remark 4.2. *If $H \subseteq \bigcup_{i=1}^{s} T_i$ (which implies $t = 0$) then* (35) *and* (36) *are identical.*

4.0.1 A facet result for pathbin inequalities

As noted in Section 2 the $CVRP$ defines a face of the $GVRP$. In contrast to the $GVRP$, which always has dimension $|E|$, the dimension of this face is not known in advance but depends on the customer demands. Most facet results for the non unit-demand case are therefore obtained only for the $GVRP$. In [CH93], Cornujoels and Harche gave sufficient conditions when facets of the $GVRP$ also define facets of the $CVRP$ but these conditions only apply when the $CVRP$ instance has the maximal dimension $|E|-|V|$. They are thus extremely restrictive and technical.

Araque has proven for his "Combs with large teeth" that they may define facets of the unit demand CVRP. In this section we will give sufficient conditions for (36) to define a facet for the general demand $GVRP$. We restrict ourselves to the case where the graph F_a has no edges, although we assume that a more general result in the spirit of Proposition 5.4 from [Ara90] can also be obtained. Our proof technique follows the proof of Theorem 3.2 in [CH93].

In order to simplify the proof as far as possible in the remainder of this section we will relax the number of vehicles, i. e. the number of nonempty tours k can take any value. This will allow us to consider partial solutions on certain "relevant" subsets of vertices without having to bother wether or not they can be extended to a k–tour for some fixed k.

As an other simplification we will assume $Z := H \setminus \bigcup_i T_i \neq \emptyset$ and $d(Z) = 0$. The second assumption is motivated by remark (4.2), i. e. it guarantees that (36) is not dominated by another pathbin inequality. In order to clarify the structure of the solutions in Figure 2, we always require some node with zero demand in Z, though. The following condition is needed for the proof.

Condition 4.3. *Let $\mathcal{C}$ be a family of node sets with a handle H and teeth $T_1, \dots, T_s$ satisfying the above conditions and for all $i \in \{1, \dots, s\}$ and all $j \in \{1, \dots, s\} \setminus \{i\}$ there exists an $l \in \{1, \dots, s\} \setminus \{i, j\}$ and a nonempty set $T_i^j \subset T_i \setminus H$ such that $d(T_j) + d\left(T_i^j\right) < C$ and $d(T_l) + d\left(T_i \setminus T_i^j\right) < C$.*

This postulates the existence of tours as depicted in Figure 2 4, 6–9.

Here the customers of $V[\mathcal{C}]$ are served by $s-1$ vehicles. With the exception of one tooth (T_i) all other teeth are served by exactly one vehicle (Note that no two teeth can be completely served by one vehicle since we assume the graph F_a to contain no edges). The customers of T_i are split among two tours, each also serving one of the "tight" teeth (T_j and T_l). Such a constellation must exist for every i and j, while l can be chosen depending on i, j. It is additionally required that the customers of $(T_i \cap H)$ are completely contained in the tour serving the costumers from T_l.

Theorem 4.4. *Let $\mathcal{C}$ be a family of nodesets with one handle and teeth $T_1, \dots, T_s$ as defined above and satisfying condition 4.3, $s \geq 3$ and*

$$d(T_i \cup T_j) > C \qquad \forall i \neq j \in \{1, \dots, s\}. \tag{37}$$

Then

$$a(\mathcal{C})x := x(\delta(H)) + \sum_{i=1}^{s} x(\delta(T_i)) \geq 4s$$

defines a facet of the $GVRP$–polyhedron.

Proof. From 4.1 it follows that $a(\mathcal{C})x \geq 4s$ is valid for the $GVRP$, when F_a contains no edge. This is implied by (37).

The remainder of the proof can be outlined as follows. First, we construct two types of solutions which show that $a(\mathcal{C})x \geq 4s$ is a face of the $GVRP$. Then we assume $fx \geq 4s$ to be a facet of $GVRP$ containing the face $a(\mathcal{C})x \geq 4s$. It is shown that the zeroes in $a(\mathcal{C})$ must also be zero in f. Next, we prove that any two edges connecting two given sets adjacent in Figure 1 must have the same coefficient in f. This reduces the number of linearly independent solutions required to uniquely determine all coefficients in f for edges between "adjacent" sets to $3s+1$. Starting from the two initial solutions a set of $3s+1$ linear

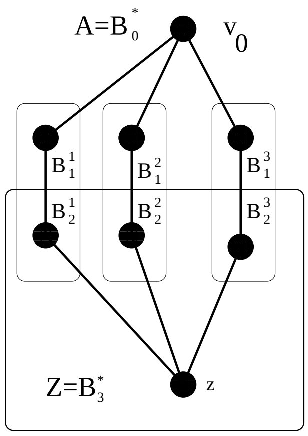

Figure 1: The sets in Theorem 4.4

independent solutions tight for $a(\mathcal{C})x \geq 4s$ is constructed. These $3s+1$ only use edges where $a_e(\mathcal{C}) = 1$ (the edges between "adjacent" sets) and can thus be used to show that $f_e = 1$ whenever $a_e(\mathcal{C}) = 1$. The solutions needed to show the same for the coeffients with $a_e(\mathcal{C}) > 1$ are not defined explicitely, here, but depicted in Figure 2, instead.

For $h = 1, \dots, s$ we define $A := V \setminus V[\mathcal{C}], B_1^h := T_h \setminus H, B_2^h := T_h \cap H, Z := H \setminus \bigcup_h T_h$ and identify $B_0^h = A$ as well as $B_3^h = Z$. The sets A, B_1^h, B_2^h, Z form a partition of the node set. Let $z \in Z$ and $b_j^h \in B_j^h$ for $h = 1, \dots, s$ and $j = 1, 2$ and $b_0^h = v_0$ and $b_3^h = z$. The sets are depicted in Figure 1. All solutions defined in the following are sketched in Figure 2. Here every set is represented by a single node, all other nodes are connected by edges with value two in the solution to this representing node. We consider first Solution 1 from Figure 2.

$$x_e^1 = \begin{cases} 2 & \text{if } e = b_j^h u \text{ with } h = 1, \dots, s; j = 0, \dots, 3; u \in B_j^h \setminus \{b_j^h\} \\ 1 & \text{if } e = b_j^h b_{j+1}^h \text{ with } h = 1, \dots, s; j = 0, \dots, 2 \\ s & \text{if } e = v_0 z \\ 0 & else. \end{cases}$$

This solution is feasible since $d(T_i) \leq C$.

Our next set of s tours utilizes condition 4.3. Namely, for $i = 1, \dots s$ we consider the solutions

$$x_e^2(i) = \begin{cases} 2 & \text{if } e = b_j^h u \text{ with } h = 1, \dots, s; j = 0, \dots, 3; u \in B_j^h \setminus \{b_j^h\} \\ 2 & \text{if } e = b_j^i b_{j+1}^i \text{ with } j = 0, \dots, 2 \\ 1 & \text{if } e = b_j^h b_{j+1}^h \text{ with } h = 1, \dots, s; h \neq i; j = 0, \dots, 2 \\ s-3 & \text{if } e = v_0 z \\ 0 & else. \end{cases}$$

(Solution 4 in Fig. 2). The tooth T_i has a special role in $x^2(i)$, since it is served by the tours that are mainly concerned with the teeth T_j resp. T_l. Due to the preconditions we may assume that the tour serving T_j does not serve any costumer in $T_i \cap H$.

Both types of solutions satisfy $a(\mathcal{C})x = 4s$, i.e. $a(\mathcal{C})x \geq 4s$ defines a face of the polyhedron and it is also easy to find a solution with $a(\mathcal{C})x > 4s$. Let $fx \geq 4s$ be a facet of $GVRP$ containing the face $a(\mathcal{C})x \geq 4s$. We show that then $a(\mathcal{C})$ and f coincide in all coefficients, i.e. $a_e(\mathcal{C}) = f_e$ for all $e \in E$. All tours that will occur in the following will have the same partition of the customers, i.e. if a client is visited by a second, modified tour he is supposed to be served further on by the same tour as before.

To determine the coefficients of f first consider an "internal" edge e with both ends lying in the same class B_i^h for some $h \in 1, \ldots, s$ and $j = 0, \ldots, 3$. Clearly $x' = x^1 + 2\chi(e)$ is another feasible tour with $\chi(e)$ being the characteristic function of $\{e\}$. Equality $a(\mathcal{C})x^1 = 4s$ implies $fx^1 = 4s$. Since $a_e(\mathcal{C}) = 0$ we obtain $a(\mathcal{C})x' = 4s$, implying $fx' = 4s$ and $f_e = 0$.

Next, we show that $f_{e_1} = f_{e_2}$ for $e_1 = (b_j^h, b_{j+1}^h)$ and any other edge $e_2 = (p_j^h, p_{j+1}^h) \in (B_j^h : B_{j+1}^h)$. Let $x'' := x^1 - \chi(e_1) + \chi((b_j^h, p_j^h)) + \chi(e_2) + \chi(p_{j+1}^h, b_{j+1}^h))$. Then x'' defines a tour and using the last paragraph and $a_e(\mathcal{C})x_1 = fx_1 = fx'' = 4s$ we conclude $f_{e_1} = f_{e_2}$. A similar argument applies to $e_1 = (v_0, z)$ and $e_2 \in (A : Z)$.

Now we consider the edges that connect sets "adjacent" in Figure 1, i.e. the $3s$ sets $(B_j^h : B_{j+1}^h)$ and $(A : Z)$. From the two preceding two paragraphs we conclude that there are at most $3s + 1$ different coefficients for the variables corresponding to these edges in $fx \geq 4s$. We will present $3s + 1$ linearly independent tours using only such edges and "internal" edges with $a(\mathcal{C})_e = 0$. The first $s + 1$ solutions are x_1 and the s linearly independent solutions $x^2(i)$ for the s choices of i. Since no customer from $T_i \cap H$ is served by the tour responsible for T_j we get another s linearly independent solutions, namely $x^3(i) := x^2(i) - \chi((b_2^i, b_3^i)) - \chi((b_1^i, b_2^i)) + \chi((b_0^i, b_1^i)) + \chi((v_0, z)$ (Solution 2 in Figure 2). Finally, we derive another s independent solutions $x^4(i)$ from x^1 for $i = 1, \ldots, s$ by setting $x^4(i) := x_1 + \chi(b_0^i, b_1^i)) + \chi(b_1^i, b_2^i)) - \chi((b_2^i, b_3^i)) - \chi((v_0, z))$ (Solution 3).

As all these solutions satisfy $a(\mathcal{C})x = 4s$ we also have $fx = 4s$. Thus the $3s+1$ free coefficients of f meeting the support of our x^j form a solution of the system $Af = b$ of linear equations. The $(3s+1 \times 3s+1)$-matrix of this system is given as

$$A := \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 & 1 & s \\
2 & 2 & 2 & 1 & 1 & 1 & \ldots & 1 & 1 & 1 & s-3 \\
1 & 1 & 1 & 2 & 2 & 2 & \ldots & 1 & 1 & 1 & s-3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
1 & 1 & 1 & 1 & 1 & 1 & \ldots & 2 & 2 & 2 & s-3 \\
3 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 & 1 & s-2 \\
1 & 1 & 1 & 3 & 1 & 1 & \ldots & 1 & 1 & 1 & s-2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
1 & 1 & 1 & 1 & 1 & 1 & \ldots & 3 & 1 & 1 & s-2 \\
2 & 2 & 0 & 1 & 1 & 1 & \ldots & 1 & 1 & 1 & s-1 \\
1 & 1 & 1 & 2 & 2 & 0 & \ldots & 1 & 1 & 1 & s-1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
1 & 1 & 1 & 1 & 1 & 1 & \ldots & 2 & 2 & 0 & s-1
\end{pmatrix}.$$

It can be verified that the absolute value of the determinant of this matrix is $4s2^{2s}$. Hence, the solution is unique and coincides with the coefficients in $a(\mathcal{C})$.

We are left with the coefficients $a(\mathcal{C})_e$ with $a(\mathcal{C})_e \geq 2$. For the proof we refer the reader to solutions $5 - 9$ in Figure 2. Each of these can be obtained from either x^1 (Solution 5) or x^2 (Solutions 6-9) without modifying the partition of V_0. They are all linearly independent since each of them owns one edge exclusively. As above we conclude that $f_e = a(\mathcal{C})_e$. $\square$

The choice of basic solutions was driven by the attempt not to use a tour serving two of the sets $T_i \setminus H$ completely. It adresses the case where all teeth have demand $> \frac{1}{2}C$ and this demand is located mainly "outside" of the handle. Unfortunately, our attemps to develop a useful separation heuristic for these "combs with large teeth" have failed.

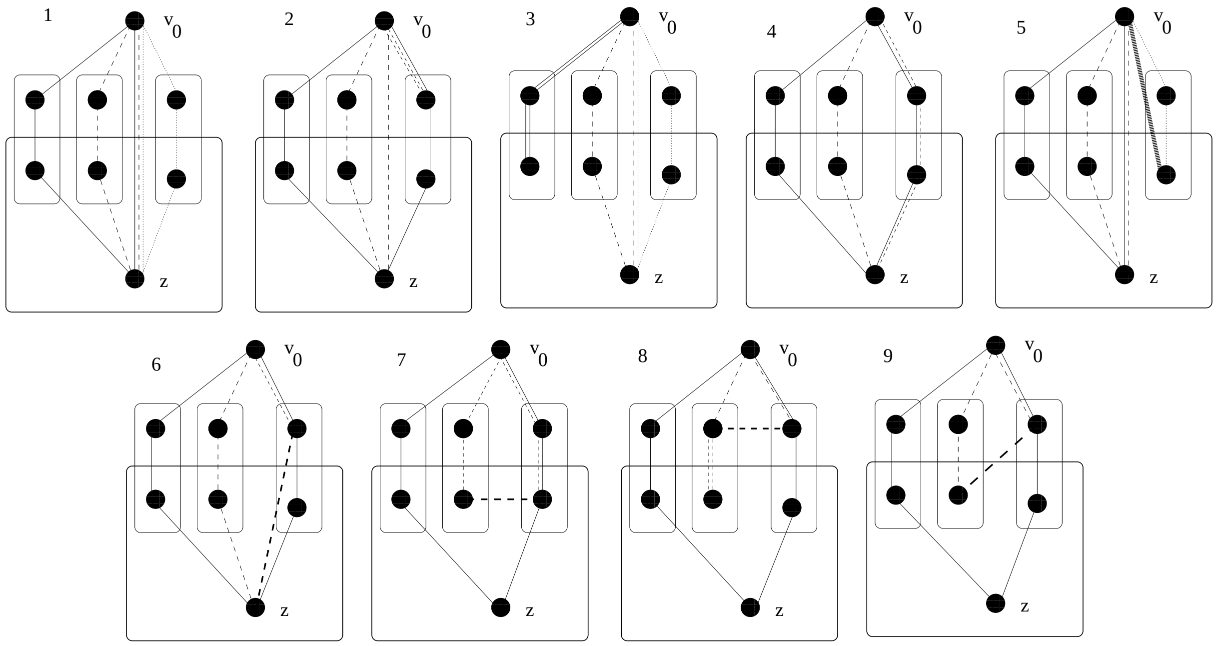

Figure 2: Basic solutions from the proof of Theorem 4.4

5 Hypotours

We will now leave our more theoretical considerations and turn to the specifics of a Branch and Cut Code for the VRP. The hypotour inequalities were introduced by Augerat and were already used in the cutting plane code presented in [ABB+95]. In our computational experiments we could confirm their finding that this class can in some cases significantly improve the lower bounds. In this section we will review the inequalities and suggest an improved separation heuristic. We will also state a method of combining hypotour inequalities to new valid inequalities.

5.1 The inequalities

Let $ax \leq a_0$ be a valid inequality with integer coefficient vector a, and let F be an edge set such that

$$ax = a_0 \Rightarrow x(F) \geq 1. \tag{38}$$

Then the hypotour inequality

$$ax - x(F) \leq a_0 - 1$$

is clearly valid for the CVRP.

We will apply this to the following set of inequalities. Let $L \subset V_0$ be a node set and $e_1 = (u_1, v_1), e_2 = (u_2, v_2) \in \gamma(L)$ and consider the valid inequalities

$$x(\gamma(L)) \leq |L| - 1 \tag{39}$$

$$x(\gamma(L \setminus \{v_1\})) + x_{e_1} \leq |L| - 1 \tag{40}$$

$$x(\gamma(L \setminus \{v_1, v_2\})) + x_{e_1} + x_{e_2} \leq |L| - 1. \tag{41}$$

In the following we will implicitly assume that a hypotour inequality is derived from one of these particular inequalities. We denote the node set L as the *central set* of a hypotour inequality and call a corresponding edge set F satisfying (38) the *completing set* of a hypotour inequality with central set L.

The practically most important subclass of hypotour inequalities arises from (41). Any k–tour satisfying this with equality must visit L in a single path starting and ending in e_1, e_2. Now, if F is a set of edges such that any k–tour containing such a path has also to pass some edge of F then

$$x(\gamma(L \setminus \{v_1, v_2\})) + x_{e_1} + x_{e_2} - x(F) \leq |L| - 2. \tag{42}$$

is valid.

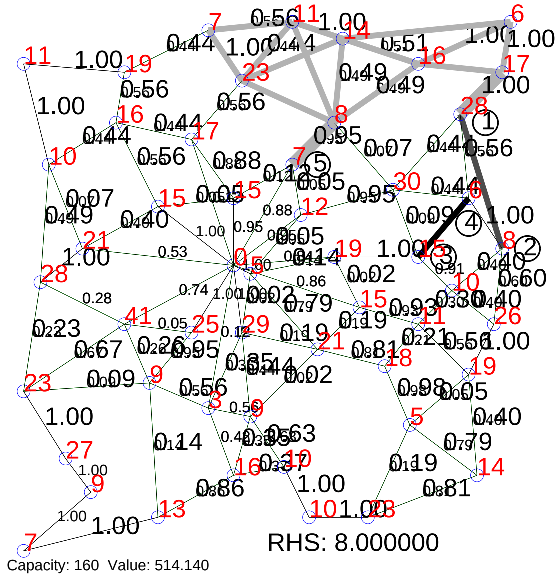

Figure 3: Hypotour inequality of G.

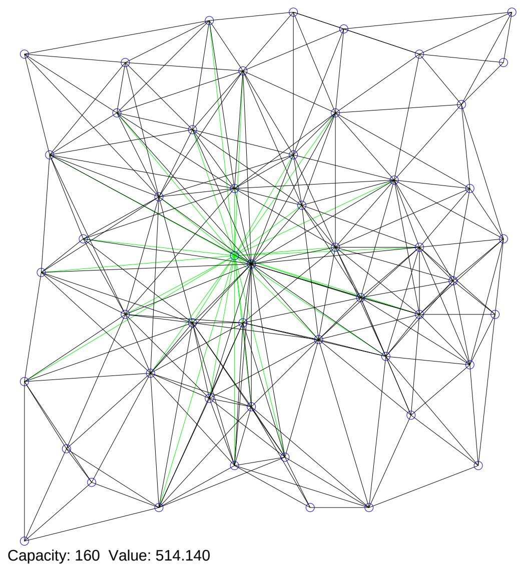

Figure 4: Not yet fixed edges of G.

For practical purposes the hypotour inequalities need not be valid for the full CVRP polytope. Often it is possible to construct completing sets F such that the resulting inequalities are guaranteed to be valid at least for the optimal face of the polytope. It is thus possible to exploit knowledge about variables which are necessarily zero in any optimal solution.

Figure 3 shows a fractional solution x in the benchmark instance E-n51-k5 [CE69], the numbers at the vertices indicate their demand. Now, let L be the set of vertices connected by the grey edges in the upper right corner and e_1, e_2 the edges emerging at their bottom. All edges not displayed in Figure 4 have already been proven to be necessarily zero in any optimal solution (the edges are fixed to zero). The capacity of this instance is $C = 160$ and $d(L) = 137$. A path through L can be completed to a tour by adding the edge $(v_0, 5)$ and either the path $(1, 2, 3, v_0)$ (with $d(L \cup \{3, 4\}) = 158$) or the path $(1, 4, 3, v_0)$ (with $d(L \cup \{2, 3\}) = 160$). The two black edges thus form a feasible completing set.

This yields a hypotour inequality of type (42) which is violated by x. The inequality shown has been identified with the separation routine presented next. The example illustrates that the choice of F is in no way unique, as there are other two element sets which could be used as a completing set. (Even the one element set containing only $(v_0, 3)$ is feasible). It also becomes evident that hypotour inequalities can be considerably stronger (i.e. F smaller) when the number of edges already fixed to zero increases.

5.2 Separation

Our separation heuristic to identify violated inequalities of type (42) has two phases. First, we enumerate candidate central sets L while in the second phase we try to identify a completing set F.

5.2.1 Enumeration of central sets

For obvious reasons we consider only candidate sets L satisfying $x(\gamma(L)) > |L| - 2$ and $d(L) \leq C$. Furthermore, we try to restrict to sets L that satisfy $x(\delta(L)) < x(\delta(L'))$ for all L' with $L \subset L'$. A hypotour inequality derived from (39) for L' dominates one for L. This is not always the case for inequalities derived from (41) but it seems reasonable to neglect this. Thus we start our enumeration by shrinking all maximal node sets S_i satisfying $x(\delta(S_i)) = 2$. The inequality $x(\gamma(L)) > |L| - 2$ implies that we only have to consider node sets L inducing a connected subgraph in the shrunken support graph of x. Furthermore, following the dominance consideration, given a node set L and a node $v \in V_0 \setminus L$ such that $x(L : v) \geq 1$ we replace L by $L \cup \{v\}$.

Computing all connected subgraphs of G_x for which no such node exists can be done quickly for the CVRP instances that we used as benchmarks, since the shrunken support graph is sparse and not too large. We used a search tree in which each tree node is associated with a current candidate set $L \subset V_0$ and a set $\overline{L} \subset V \setminus L$ of customers which can not be included into L in the current subtree. $\overline{L}$ is initialized with the depot. In each enumeration node a customer $v \in V \setminus (L \cup \overline{L})$ is chosen and two sons with $v \in L$ and $v \in \overline{L}$ are created. A node in the search tree is eliminated if one of the following conditions is met

$$\begin{aligned} d(L) &> C \\ x(L : \overline{L}) &\geq 4 \\ \exists_{v \in \overline{L} \setminus \{v_0\}} x(v : L) &\geq 1 \\ x(L : v_0) &\geq 2. \end{aligned}$$

In the second case $x(\gamma(L')) \leq |L'| - 2$ for any set L' which can be generated in the current subtree and thus L' cannot serve as a central set. In the third case each set L' will be dominated by some other set containing the node v. If the last condition is met there is hardly a chance to find a completing set F for any L' contained in the current subtree. If this enumeration is implemented carefully the number of candidate sets L is sufficiently small and in phase two we check for all candidate sets satisfying $d(L) > 0.6 \cdot C$, whether a violated hypotour inequality can be constructed. Table 2 shows the number of generated candidate sets in some benchmark instances and reveals that the time spent in this enumeration can be neglected.

5.2.2 Computing the completing set

The difficult and time consuming part of the heuristic is the computation of a small completing set F. We only consider the case of inequality (42). Here, F has to contain at least one edge from every feasible k–tour that visits the nodes in L in a Hamiltonian path visiting L with edge set ending in e_1, e_2.

Again we try to keep F small by relaxing the requirement in (38) to incidence vectors of optimal k–tours only. The resulting hypotour inequality is still valid for the optimal face of the polytope. This is done enumeratively by first determinining a short Hamiltonian path and then completing it to a tour T. Within the enumeration we check whether T can itself be improved by a two-opt step, or by moving a single node to another position within this tour. If no local improvement can be found we add an edge from T with sufficiently small x–value to F. If no such edge exists, then in a third step we apply the following criterion:

Restcluster criterion: *Let G_x denote the support graph of x and let $C_1, \ldots, C_s$ be the components of $G \setminus (\{v_0\} \cup T)$. If there is a maximal connected component C_i with $d(C_i) < d(V_0) - (k-1)C$, then add all edges of $(C_i : V_0 \setminus (C_i \cup T))$ to F.*

This criterion yields a correct completing set F, as there can not be a k–tour containing a tour with a demand less than $d(V_0) - (k-1)C$. Thus there must be some edge from $(C_i : V_0 \setminus (C_i \cup T))$ in any k–tour containing tour T.

Name	avg. #enumeration node	avg. #tested central sets	avg. time sep.-heur.[s]	time tests [%]
B-n57-k9	629	41	0.07	77.5
E-n101-k8	1351	77	1.3	97.6
E-n51-k5	232	28	0.05	88.9
E-n76-k10	826	85	0.6	94.4
FRB-n51-k5	506	29	0.21	94.9
F-n135-k7	1820	49	12.8	99.5

Table 2: Performance of the enumeration of central sets

Name	Upper Bound	LB-Hypos	Num. Iter.	Time [s] (Separation [s])	Time [s] Hypotour-Sep.	LB No Hypos
B-n57-k7	1153	1149.68	75	22 (16.8)	1.9	1148.68
B-n57-k9	1598	1587.59	94	33.5 (24.3)	1.5	1586.44
B-n63-k10	1496	1479.98	71	20.38 (13.3)	0.8	1478.85
E-n101-k8	817	800.178	74	118 (104)	36.4	796.341
E-n30-k3	534	509.5	19	0.7 (0.5)	0.3	508.5
E-n33-k4	835	> 834	18	0.7 (0.4)	0.03	833.5
E-n51-k5	521	517.8	48	7 (5.7)	0.5	514.524
E-n76-k10	835	794.154	85	86 (71)	15.4	788.789
E-n76-k7	682	663.705	62	41.3 (35.4)	5.8	661.247
E-n76-k8	735	713.153	70	53.6 (44.8)	6.2	710.91
FRA-n51-k5	2552	2545.69	57	13.1 (10.8)	1.1	2541
FRB-n51-k5	4195	4139.33	54	12.5 (8)	1	4136
FRC-n51-k5	3772	3701.85	74	28.7 (23)	4.3	3688.71
F-n72-k4	237	232.558	25	5 (3.9)	2.1	232.5
F-n135-k7	1162	1159.25	118	256 (170)	40.7	1156.87

Table 3: Results of cutting plane code with hypotour separation

Summarizing we do the following. Given a set L we compute the optimal path through L ending in e_1, e_2 and then do a depth first type enumeration of all possible extensions of this path to a complete tour. The optimality tests for this tour can be done in time linear in the current length of T whenever a new node is appended within the search and thus help to keep the enumeration tree small. Whenever a feasible tour T is found, we try to include edges either from T with sufficiently small x-value or satisfying the Restcluster criterion into F. If this fails we continue with the next central set. Otherwise, in the following, tours which use this edge need no longer be considered. Still, the enumeration can lead to combinatorial explosion and we had to introduce some stopping criteria.

5.2.3 Results

Table 3 summarizes the results of the separation heuristic for hypotour inequalities. We used in the runs all heuristics already described in [ABB+95] to identify capacity constraints and resorted to the hypotour separator only when not enough (at least 5) capacity constraints were found. The last column contains the results we obtained when only capacity constraints where used as cutting planes.

5.3 Combination of Hypotours

In this section we describe a class of valid inequalities which can be obtained from the combination of the hypotour inequalities resulting from (39)-(41) as described in the previous section. The underlying observation is that if a solution contains two node disjoint paths P_1, P_2 with a total demand of more than C then P_1 and P_2 belong to different tours and these two tours have no edge in common.

Let $x(E_{L_1}) - x(F_1) \leq |L_1| - 2$ and $x(E_{L_2}) - x(F_2) \leq |L_2| - 2$ be valid hypotour inequalities with central sets L_1 and L_2 and completing sets F_1, F_2. For simplicity we assume $L_1 \cap L_2 = \emptyset$.

Definition 5.1. *Let F be a completing set for a central set L. Let kT be a feasible k–tour that visits L in a single path and intersects F in a single element e. Call such an edge e* exclusive *if there is no k–tour visiting L in a single path but e in a different tour. We call a set $F^{tour} \subseteq F$ of exclusive edges* exclusive, *as well as the partition (F^{tour}, F^{notour}) where $F^{notour} := F \setminus F^{tour}$.*

Lemma 5.2. *Given two valid hypotour inequalities as described above with disjoint central sets L_1, L_2 such that $d(L_1 \cup L_2) > C$ and completing sets F_1, F_2, let F_i^{tour}, F_i^{notour} be exclusive partitions of F_i for $i = 1, 2$. Then the combined hypotour inequality*

$$x(E_{L_1}) + x(E_{L_2}) - x(F_1^{tour} \cup F_2^{tour}) - x(F_1^{notour}) - x(F_2^{notour}) \leq |L_1| + |L_2| - 4 \tag{43}$$

is valid.

Proof. Let x be the incidence vector of a k–tour kT. Two subtour elimination constraints imply $x(E_{L_1}) + x(E_{L_2}) \leq |L_1| + |L_2| - 2$. Thus it suffices to examine the following two cases.

First assume $x(E_{L_1}) + x(E_{L_2}) = |L_1| + |L_2| - 3$. Then one of $x(E_{L_i}) \leq |L_i| - 1$ for $i = 1, 2$ must be satisfied with equality and (43) is obtained as a sum of the corresponding hypotour inequality and the subtour elimination constraint for L_j, $j \neq i$.

If $x(E_{L_1}) + x(E_{L_2}) = |L_1| + |L_2| - 2$, we have $x(F_i) \geq 1$ for $i = 1, 2$. If $|kT \cap (F_1 \cup F_2)| \geq 2$ we are done. Thus we may assume $kT \cap F_1 = kT \cap F_2 = \{e\}$. Still we are done unless $e \in F_1^{tour} \cap F_2^{tour}$. Exclusiveness of these sets implies that $L_1 \cup L_2$ are served by a single tour, a contradiction. □

Remark 5.3. *The combined hypotour inequality dominates the sum of the two original inequalities iff*

$$(F_1^{tour} \cap F_2^{tour}) \setminus (F_1^{notour} \cup F_2^{notour}) \neq \emptyset. \tag{44}$$

A recursive application of this method yields combinations also of $l > 2$ hypotour inequalities:

$$\sum_{i=1}^{l} \left(x(E_{L_i}) - x(F_i^{notour}) \right) - x\left(\bigcup_{i=1}^{l} F_i^{tour}\right) \leq \sum_{i=1}^{l} |L_i| - 2l.$$

An example of the combination of three inequalities is shown in Figure 5. The thick horizontal edge in the center belongs to F_1^{notour} and F_2^{notour}.

In practice an exclusive partition can be obtained within the enumerative procedure described in the previous section by including into F_i^{notour} every edge, which has been added to F_i due to the restcluster criterion.

In the cutting plane code described below we used combined hypotours which we obtained from simple hypotours generated in previous cutting-plane iterations by checking (44). The improvement of the lower bounds using these inequalities was very small. Nevertheless the check can be done very quickly.

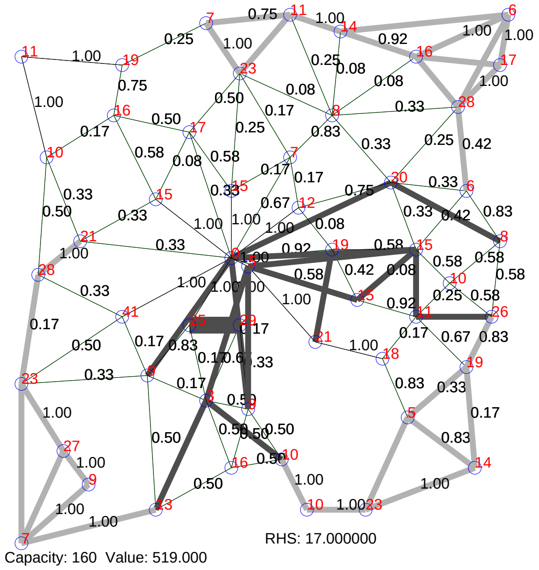

Figure 5: Combination of hypotour inequalities.

6 Branch and Cut

We implemented a Branch and Cut code for the symmetric CVRP based on the ABACUS-library by Thienel and Jünger [JT98] with CPLEX 6.0 as LP-solver. Table 4 summarizes the results we obtained with our implementation of a cutting plane code for the CVRP. We used all separators described in [ABB+95] for the identification of violated capacity constraints and a simplified version of their algorithm for generalized capacity constraints. Additionally, we used a heuristic which tries to find a violated capacity constraint by greedily adding nodes to the central set of a violated multistar constraint, which was found by the network flow algorithm described in Section 3. Furthermore, we identified violated comb inequalities with a routine similar to the one described in [ABB+95]. Some modifications of this routine were introduced in order to find more combs containing the depot in one of the teeth. We also implemented the Padberg–Rao [PR82] algorithm for 2–matching inequalities. This routine is exact as long as all demands are smaller than $d(V) - (k-1)C$. In the other cases the routine may find only inequalities which are not valid for the CVRP but this can easily be checked. We also used the separation routine for hypotours from Section 5.

The runs were done on a 400 MHz Sun–Ultrasparc II and are in the same order of magnitude as those from [ABB+95], taking into account the differences in CPU speed and quality of the CPLEX version. An exception is the instance F-n135-k7, where we needed several times longer. This time is spent mainly for hypotour separation, indicating that the stopping criteria should perhaps be adapted during running-time in order to obtain a better scalability of the algorithm. We stop as soon as the gap is smaller than which is why for some instances we have no exact values for the lower bound. In these runs the upper bounds were passed to the program as external parameters.

It can be seen that the practical difficulty in solving a given instance of the CVRP to optimality increases when the tours in the optimal solution tend to intersect geographically. This is forced by small vehicle capacities, as can be seen by comparing the results of E-n76-k7, E-n76-k8 and E-n76-k10. These are identical except for the value of C. Another feature of an instance which destroys geographical clustering in the solution is when the depot is not in the center of the customers. This can be seen for example from the instances FRA-n51-k5, FRB-n51-k5 and FRC-n51-k5. These are identical except that in FRB

Name	Upper Bound	Lower Bound	Iter.	time [s]	time [s] separation	LB [ABB+95]
B-n57-k7	1153	1149.64	67	18.3	13.6	
B-n57-k9	1598	1587.99	98	40.2	30.5	
B-n63-k10	1496	1480	74	23.1	15.4	
E-n101-k8	817	800.97	134	257	234.8	799.656
E-n22-k4	375	375	8	0.1	0.05	375
E-n23-k3	569	569	7	0.1	0.05	569
E-n30-k3	534	513.2	36	1.6	1.4	534
E-n33-k4	835	$\geq$ 834	18	0.7	0.4	835
E-n51-k5	521	518.623	70	17.3	15.5	517.124
E-n76-k10	835	795.17	172	290	262	793.384
E-n76-k7	682	664.704	105	95	87	664.355
E-n76-k8	735	714.7	125	126	112	713.746
FRA-n51-k5	2552	2546.07	64	18.6	16.1	
FRB-n51-k5	4195	4140.09	66	21.3	16.2	
FRC-n51-k5	3772	3703.68	111	63.6	56.4	
F-n45-k4	724	>723	26	1.2	0.7	724
F-n72-k4	237	234.143	63	16.2	13.8	235
F-n135-k7	1162	1159.56	155	666	572	1159.06
M-n101-k10	820	>819	39	15.1	6.7	820

Table 4: Cutting plane results

and FRC the depot is located somewhere at the border while in FRA there is a "central" depot.

All runs were initialized with an upper bound of optimal value plus one. An exception was the instance E-n76-k8 where we used the optimal value as the initial upper bound, since our heuristics failed to find the optimal solution quickly enough. Upper bounds were computed during the runs with a tabu-search and a savings heuristic with an edge weight function where the distances were modified using the current LP-solution. In Table 5 we summarize the results of our Branch and Cut implementation for those instances which could be solved to optimality.

Much effort is spent in the branching step. We used branching on sets as already proposed by other authors (cf. [ABB+95]). Given a set L with $x(\delta(L)) \approx 3$ we generated two sons with $x(\delta(L)) = 2$ and $x(\delta(L)) \geq 4$. In order to evaluate the quality of a set L we solved the resulting linear program for the "$x(\delta(L)) \geq 4$"–branch. We tested up to 20 such sets per branching-step and chose the one for which the "$x(\delta(L)) \geq 4$"–branch lead to the largest increase in the objective. This strategy was due to the observation that the branching strategy tends to produce unbalanced enumeration trees with the "≥ 4"–side having significantly more nodes than the "$x(\delta(L)) = 2$"–side where sets of customers are shrunk to nodes with large demands. We also solved the "$x(\delta(L)) = 2$"–LP for all candidate branching sets. Whenever the resulting lower bounds in one of the two tested branches already revealed that no improvement of the upper bound could be achieved in the corresponding subtree we added the branching–inequality to the current LP and reentered the cutting plane phase in the current node. With this strategy the instance FRA-n51-k5 could be solved within the root node even though the cutting plane procedure itself could not really close the gap (see Table 4).

Within the non–root subproblems we did not use the separation routines for combs any more, because they did not contribute significantly to an improvement of the lower bound. It seems remarkable that with the instances E-n76-k7 and E-n76-k8 two of the 76-node instances published by Eilon [CE69] were solved for the first time on a single processor. Independently from our work, Kopman, Ralphs, Pulleyblank and Trotter [Kop99, KRPT00, NR01] solved these instances on a parallel machine using a huge enumeration tree. An optimal solution with cost 682 of instance E-n76-k7 is depicted in Figure 6. Nevertheless the

Name	Optimal value	B&B nodes	Time [h:min:s]	Time [h:min:s] Primal Heuristics	Time [h:min:s] Branching Tests
B-n57-k7	1153	103	0:11:01	0:08:13	0:01:00
B-n57-k9	1598	493	0:39:48	0:20:13	0:08:23
B-n63-k10	1496	1581	0:32:40	0:09:07	0:06:51
FRA-n51-k5	2552	1	0:02:00	0:01:43	0:00:01
FRB-n51-k5	4195	463	0:17:12	0:07:10	0:03:32
FRC-n51-k5	3772	827	0:27:45	0:03:59	0:09:26
F-n72-k4	237	41	0:01:15	0:00:33	0:00:03
F-n135-k7	1162	65	1:31:29	1:18:34	0:03:21
E-n76-k7	682	6717	7:39:10	1:04:22	1:57:45
E-n76-k8	$735^{(a)}$	6259	9:51:06	4:50:10	2:55:38

Table 5: Branch and Cut results. $^{(a)}$Upper bound 735 was provided as external parameter.

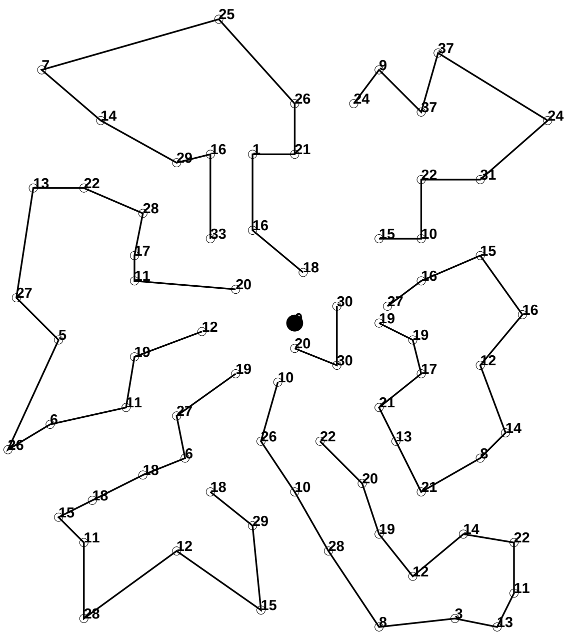

Figure 6: Optimal solution of instance E-n76-k7 with value 682.

challenging instance E-n76-k10 still resisted all attacks.

Acknowledgement. The authors are grateful to three anonymous referees who made several valuable suggestions improving a preliminary version of the paper and to Denis Naddef for bringing the work of Augerat to their attention.

References

[ABB+95] P. Augerat, J. M. Belenguer, E. Benavent, A. Coberan, D. Naddef, and G. Rinaldi. Computational results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem. Rapport de recherche RR949-M, 1995.

[AHM90] J. R. Araque G., L. A. Hall, and T. L. Magnanti. Capacitated Trees, Capacitated Routing and Associated Polyhedra. CORE Discussion Paper No. 9061, 1990.

[AMO93] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows.* Prentice Hall, 1993.

[Ara90] J. R. Araque G. Lots of combs of different sizes for vehicle routing. CORE Discussion Paper, 1990.

[Aug] P. Augerat. Vrp-instances. http://www-apache.imag.fr/-paugerat/VRP/INSTANCES.

[Bla00] U. Blasum. *Anwendung des Branch & Cut Verfahrens auf das kapazitierte Vehicle-Routing Problem.* PhD thesis, Universität zu Köln, Cologne, Germany, 2000.

[BMMN95] M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser. Network routing: Handbooks in operations research and management science. *Elsevier Science, Amsterdam*, 8, 1995.

[CE69] N. Christofides and S. Eilon. An algorithm for the vehicle dispatching problem. *Operations Research Quaterly*, 20:75–85, 1969.

[CFN85] G. Cornuéjols, J. Fonlupt, and D. Naddef. The Travelling Salesman Problem on a Graph and some related Integer Polyhedra. *Math. Prog.*, 33:1–27, 1985.

[CH93] G. Cornuéjols and F. Harche. Polyhedral study of the capacitated vehicle routing problem. *Mathematical Programming*, 60:21–52, 1993.

[FJ81] M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11:109, 1981.

[Gav84] B. Gavish. The delivery problem: New cutting planes procedure. presented at the TIMS XXVI conference, Copenhagen, 1984.

[Gou95] L. Gouveia. A result on projection for the vehicle routing problem. *Europ. Journal of Op. Res.*, 85:610–624, 1995.

[HCM95] E. Hadjiconstantinou, N. Christofides, and A. Mingozzi. A new exact algorithm for the vehicle routing problem based on q-paths and k-shortest path relaxations. *Annals of Op. Res.*, 61:21–43, 1995.

[JRR95] M. Jünger, G. Reinelt, and G. Rinaldi. The traveling salesman problem. In M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science.* North-Holland, Amsterdam, 1995.

[JT98] M. Jünger and S. Thienel. Introduction to abacus – a branch-and-cut system. *Operations Research Letters*, 22(2-3):83–95, 1998.

[Kop99] L. Kopman. *A New Generic Separation Routine and its Application in a Branch and Cut Algorithm for the Capacitated Vehicle Routing Problem.* PhD thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaka, 1999.

[KRPT00] L. Kopman, T. Ralphs, W.R. Pulleyblank, and L.E. Trotter. A parallel branch and cut algorithm for vrp. in preparation, 2000.

[Lap92] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59:345–358, 1992.

[LE96] A. N. Letchford and R. W. Eglese. New cutting-planes for vehicle routing problems. Presented at CO96, London, 1996.

[LN84] G. Laporte and Y. Norbert. Comb inequalities for the vehicle routing problem. *Methods of Op. Res.*, 51:271–276, 1984.

[MT90] Silvano Martello and Paolo Toth. *Knapsack Problems.* John Wiley & Sons, 1990.

[NR01] D. Naddef and G. Rinaldi. Branch and cut. In P. Toth and D. Vigo, editors, *Vehicle Routing*, chapter 3. SIAM, 2001.

[PR82] M. Padberg and M. R. Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7:67–80, 1982.

[Rei91] G. Reinelt. A traveling salesman problem library. *ORSA Journal on Computing*, 3:376–384, 1991.

[YKH89] Y. Agarwal, K. Mathur, and H.M. Salkin. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks*, 19:731 – 749, 1989.