Minimum Cut Bases in Undirected Networks

Florentine Bunke*, Horst W. Hamacher*, Francesco Maffioli†, Anne Schwahn*

March 23, 2007

Abstract

Given an undirected, connected network $G = (V, E)$ with weights on the edges, the cut basis problem is asking for a maximal number of linear independent cuts such that the sum of the cut weights is minimized. Surprisingly, this problem has not attained as much attention as its graph theoretic counterpart, the cycle basis problem. We consider two versions of the problem, the unconstrained and the fundamental cut basis problem.
For the unconstrained case, where the cuts in the basis can be of an arbitrary kind, the problem can be written as a multiterminal network flow problem and is thus solvable in strongly polynomial time. The complexity of this algorithm improves the complexity of the best algorithms for the cycle basis problem, such that it is preferable for cycle basis problems in planar graphs. In contrast, the fundamental cut basis problem, where all cuts in the basis are obtained by deleting an edge, each, from a spanning tree T is shown to be NP-hard. We present heuristics, integer programming formulations and summarize first experiences with numerical tests.
Keywords: cut basis problem, graph and network algorithms, integer programming

*Fachbereich Mathematik, Technische Universität Kaiserslautern, Kaiserslautern, Germany; E-mail: {flobunke,hamacher,schwahn}@mathematik.uni-kl.de. The research has been partially supported by Deutsche Forschungsgemeinschaft (DFG) grant HA 1737/7 "Algorithmik großer und komplexer Netzwerke", New Zealand's Julius von Haast award, and by the Rheinland-Pfalz cluster of excellence "Dependable adaptive systems and mathematical modeling".
†Department of Electronics and Information, Politecnico di Milano, Milan, Italy; E-Mail: maffioli@elet.polimi.it

1 Introduction

Let $G = (V, E)$ be an undirected, connected, simple graph with n vertices, m edges and a weight function $w : E \longrightarrow \mathbb{R}_+$ on the edges. In G, we consider the set of all *cuts* $D = (U, V \setminus U) = (U, \overline{U})$ which is well-known to form a vector space over the two-element field $\mathbb{F}_2$, the *cut space* $\mathfrak{D}$ of G. The dimension of $\mathfrak{D}$, and thus the number of cuts in any cut basis, is $n - 1$.

The weights w_e for the edges $e \in E$, can be carried over to weights $w(D) = \sum_{e \in D} w_e$ for each cut D, and to $w(\mathcal{D}) = \sum_{i=1}^{t} w(D_i)$ for any cut basis $\mathcal{D} = \{D_1, \ldots, D_t\}$. The *min cut basis problem* MinDB consists in finding a cut basis $\mathcal{D}$ with minimum value $w(\mathcal{D})$. If we want to emphasize the fact that the weights are summed up we call the problem also *min-sum cut basis problem.* Another problem is the *min-max cut basis problem* $min\{max\ w(D) : D \in \mathcal{D}\}$. The non-negativity of the weights implies that each cut D in a minimum cut basis $\mathcal{D}$ is an elementary cut (i.e. no subset of D is disconnecting).

In addition to the unconstrained cut basis problem we may also enforce additional properties on the bases in $\mathfrak{D}$. In this paper, we focus on one constraint, the fundamentality of the cuts: Given a spanning tree of G the deletion of any edge $e \in E(G)$ separates the node set V of G into two sets U_e and $\overline{U}_e$. Thus $D_e = (U_e, \overline{U}_e)$ is a cut in G. Obviously, the set $\mathfrak{D} = \{D_e : e \in E(T)\}$ of $n - 1$ cuts is linearly independent and is thus a cut basis, a *fundamental cut basis.* In the *fundamental min cut basis problem* MinFDB we want to find a spanning tree $T = (V, E(T))$ such that its objective value is minimum among all fundamental cut basis.

The min cut basis problem is closely related to its graph theoretic counterpart, the *min cycle basis problem.* We use the denotations MinCB and MinFCB for the unconstrained and fundamental min cycle basis problem, respectively. In contrast to the cut basis problems, MinCB and MinFCB have been studied extensively in the past. The first polynomial algorithm for MinCB is due to Horton [Hor87]. Its complexity was improved by [GPS02] and [KMMP04]. Horton's idea also led to a polynomial algorithm for problem MinCB with min-max objective function in [Gal01]. Problem MinFCB, on the other hand, is known to be NP-hard even for uniformly weighted graphs as was shown in [DPK82] for the min-sum objective and in [Gal01] for the min-max case. The problems stay NP-hard even if the graph is complete with non-uniform weights, but are polynomially solvable for complete, uniform graphs [Gal01]. Several approximability results have been established for MinFCB [Gal01] and [GA03]. Integer programming formulation and metaheuristics are presented in [ALMM04]. Exact solutions can only be found for small instances of the cycle base problem. Several suggestions for constructive heuristics solving larger instances can be found in [DPK82], [DKP95].

Although the unconstrained and fundamental cut basis problems have as much potential for applications as the corresponding cycle problems (electrical networks, estimation of damage in network failure, etc.) and are of theoretical interest in their own right, there is - in contrast to the cycle problems - hardly any literature on MinDB and MinFDB. To the best of our knowledge, the only publications are [GH02], who treat MinDB as special case of a generalized base problem for regular matroids, a chapter in the Ph.D. thesis of Bunke [Bun06] and the diploma thesis of Schwahn [Sch05].

In the next section we will give some results for general cut bases and present a polynomial algorithm for MinDB. For planar graphs, this approach can be used to solve MinCB as well, thus improving current algorithms. In Section 3 we will prove that MinFDB is NP-hard, show that two special cases can be solved in polyomial time and present different heuristics for tackling MinFDB . The next section is devoted to various integer programming formulations for MinFDB . Two of these formulations use a cross product of the tree and cut base polytope, while the third uses a characterization of fundamental cut basis which does not require the tree polytope. First numerical experiences with the approaches suggested in this paper are reported in Section 5. The paper concludes with a summary of our findings and a list of current and suggestions for future work on cut basis.

2 Solving the unconstrained cut basis problem

Since linear independence is a special case of matroid independence, a min cut basis can, in principle, be found by applying a greedy algorithm: Find iteratively the best cuts until $n-1$ linearly independent cuts have been found. The generation of the cuts can be done by using the ranking procedure of [HQ85] which is applicable to general combinatorial optimization problems. This approach has the obvious drawback that, in the worst case, exponentially many cuts have to be generated before a minimum basis is found.

Nevertheless, the validity of the greedy approach implies the following result, since every min-sum cut basis can be constructed by the greedy approach and every solution derived by this approach solves the min-sum as well as the min-max problem.

Proposition 2.1. *Any solution for the unconstrained min-sum cut basis problem is also a solution for the min-max cut basis problem.*

In the special case, where G is a planar graph, it is easy to prove that min cut and min cycle bases problems are equivalent.

Proposition 2.2. *If G is a planar graph and G^* its dual graph, then the min cut basis problem in G is equivalent to the min cycle basis problem in G^*.*

From Proposition 2.2 it is clear that in planar graphs the min cut basis problem MinDB can be solved in polynomial time using any of the polynomial algorithms for solving the min cycle basis problem MinCB ([Hor87],[GPS02], [KMMP04]). In the following we will, however, show that MinDB can be solved by the Gomory-Hu algorithm [GH61] more efficiently. Its complexity improves the complexity of the best algorithms for MinCB such that it is preferable to solve MinCB in planar graphs with this algorithm. It should be noted that [GH02] already observed the applicability of the Gomory-Hu approach in the context of circuit bases for regular matroids, albeit without giving a validity proof. In order to do the latter, we start by proving some general results for cut bases which are also interesting in their own right.

Lemma 2.3. *Let $\mathcal{D}$ be a minimum cut basis of G. Then for every pair of vertices, there exists a minimum weight cut $D \in \mathcal{D}$ that separates these vertices.*

Proof. Let $\mathcal{D}$ be a minimum cut basis of G and let $u, v \in V$ be any pair of vertices in G. The basis property of $\mathcal{D}$ implies that $\mathcal{D}$ contains a cut separating u and v. Suppose that none of these cuts in $\mathcal{D}$ is of minimum weight. Then $w(D_{uv}) < w(D)$ holds for any minimum weight u-v-separating cut D_{uv} and all $D \in \mathcal{D}$. On the other hand, the basis representation of D_{uv} in terms of the cut basis $\mathcal{D}$ – as a symmetric difference of cuts in $\mathcal{D}$ – has to contain at least one u-v-separating cut D. Clearly, $\mathcal{D}' := \mathcal{D} \setminus \{D\} \cup \{D_{uv}\}$ is again a cut basis, but with weight $w(\mathcal{D}') = w(\mathcal{D}) - w(D) + w(D_{uv}) < w(\mathcal{D})$. This is a contradition to the minimality of $\mathcal{D}$. □

This result can be used to give a characterization of min cut bases using non-crossing cuts. (Two cuts $D_1 = (U, \overline{U})$ and $D_2 = (W, \overline{W})$ are called *crossing* if all four set intersections $U \cap W, U \cap \overline{W}, \overline{U} \cap W$, and $\overline{U} \cap \overline{W}$ are nonempty.)

Theorem 2.4. *Let $\mathcal{D}$ be a collection of non-crossing independent cuts of the graph G. Then the following statements are equivalent:*

a) *$\mathcal{D}$ is a minimum cut basis of G.*

b) *For every pair u, v of vertices in G, $\mathcal{D}$ contains a minimum weight u, v-separating cut and $\mathcal{D}$ is a minimal set with this property.*

Proof. Let $\mathcal{D}$ be a minimum cut basis. By Lemma 2.3, $\mathcal{D}$ is a set with the property required in b). The minimality of $\mathcal{D}$ follows from its non-crossing property which implies that every cut $D \in \mathcal{D}$ separates a pair of vertices that is separated by no other cut in $\mathcal{D}$ and thus, can not be deleted.
Conversely, we prove that $\mathcal{D}$ can be obtained as output of the Greedy Algorithm applied to the set of all cuts of G if (b) holds. In the first step, the Greedy Algorithm chooses the minimum cut among all cuts of G, and such a cut has to be also contained in $\mathcal{D}$. Assume that after iteration i, the Greedy Algorithm has already chosen a set of cuts $\mathcal{D}_i \subset \mathcal{D}$. By the non-crossing property of $\mathcal{D}_i$, the independent cut to be selected in the next iteration has to separate a new pair of vertices that is not yet separated. Moreover, every cut separating a new pair of vertices that is not yet separated can clearly not be represented as a symmetric difference of cuts in $\mathcal{D}_i$ and thus is independent of $\mathcal{D}_i$. It follows that the candidate cuts for the Greedy Algorithm are exactly the cuts of minimum weight separating a new pair of vertices. By property b), such a candidate cut can be chosen out of the set $\mathcal{D}$. □

Under the assumption of non-crossing cuts, Theorem 2.4 implies that MinDB corresponds to the problem of finding minimum cuts separating all pairs of vertices. The latter can be achieved in strongly polynomial time by solving a multi-terminal network flow problem using the Gomory-Hu algorithm [GH61] or any of its improvements (see, e.g., [Gus90]). The outcome of these algorithms is a *minimum cut tree*, in which each edge defines a cut of a cut basis. We thus get the following result.

Theorem 2.5. *The minimum cut basis problem* MinDB *can be solved by finding a minimum cut tree T. In particular, it is polynomially solvable in $\mathcal{O}(n\,\mathcal{K})$ time, where $\mathcal{K}$ is the complexity of finding a minimum u-v cut.*

Note, that the minimum cut tree T of Theorem 2.5 is, in general, not a subtree of G and thus not a solution of the fundamental cut basis problem.

Another alternative to solve MinDB in polynomial time is to generalize the algorithm of [GPS02] by iteratively finding minimum weight odd cuts in a signed graph. Since the complexity of the resulting algorithm is, however, much worse than $\mathcal{O}(n\,\mathcal{K})$, the complexity of the minimum cut tree algorithm, we do not present this algorithm but refer to [Bun06] for details.

3 Complexity of the fundamental cut basis problem and heuristics

The example of Figure 1 with edge weights equal to 1 for all edges shows that the optimal objective value of the fundamental min cut basis problem MinFDB is in general worse than the optimal objective value of its relaxation, the unconstrained minimum cut basis problem MinDB. Here, MinDB has an optimal objective value of $Z_{\text{MinDB}} = 16$, which is for example reached by the isolated nodal cut basis $\mathcal{D} = \{D(1), D(2), D(3), D(4), D(6)\}$. But the best value we can get when the cut basis is required to be fundamental is $Z_{\text{MinFDB}}(T) = 17$, which is, for instance, obtained by the fundamental basis corresponding to the spanning tree of Figure 1.

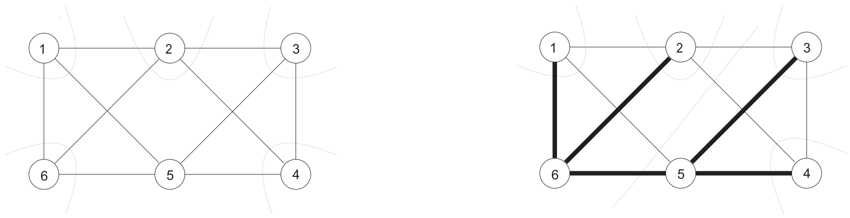

Figure 1: Optimal nodal cut basis for MinDB with weight 16 and optimal spanning tree T for MinFDB with weight 17.

The proof of our complexity result is based on the following, well-known relation between fundamental cuts and fundamental cycles.

Lemma 3.1. *Let $T = (V, E(T))$ be a spanning tree of graph G and let c be a chord of T (i.e., $c \in (E \setminus E(T))$). Furthermore let $C(T, c) = \{e_1, \dots, e_k\} \cup \{c\}$ with $e_1, \dots, e_k \in E(T)$ be the fundamental cycle with respect to c and T.*
Then c is contained in a fundamental cut $D(T, e)$ defined by T and $e \in E(T)$ if and only if $e = e_i$ for some $i = 1, \dots, k$.

Note that analogously, a branch $e \in E(T)$ that determines the fundamental cut $D(T, e)$ belongs to precisely all those fundamental cycles which are determined by the chords of T contained in $D(T, e)$. Using Lemma 3.1 we can establish a relation between the objective functions of MINFDB and MINFCB for uniformly weighted graphs thus proving the following result.

Theorem 3.2. *The minimum fundamental cut basis problem* MINFDB *is NP-hard even in the case of uniform edge weights.*

Proof. The decision version of MINFCB , i.e. the problem whether there exists a spanning tree T for which $z_{\text{MINFCB}}(T) \leq k$, is known to be NP-complete even when all weights are one (see [DPK82]). By Lemma 3.1, the objective function $z_{\text{MINFDB}}(T)$ of MINFDB can be expressed in terms of fundamental cycles as

$$z_{\text{MINFDB}}(T) = \sum_{e \in T} w_e + \sum_{e \notin T} (|C(T, e)| - 1) \cdot w_e, \tag{3.1}$$

where $|C(T, e)|$ denotes the number of edges in the fundamental cycle $C(T, e)$. In the case where all edge weights w_e are equal (w.l.o.g. equal to one), this relation reduces to

$$\begin{aligned} z_{\text{MINFDB}}(T) &= n - 1 + \sum_{e \notin T} |C(T, e)| - (m - n + 1) \\ &= 2n - m - 2 + z_{\text{MINFCB}}(T). \end{aligned}$$

Hence both objective functions of MINFDB and MINFCB just differ by a constant and an optimal tree for one of the two problems is also optimal for the other.

□

As the following results show, there are, however, special cases of MINFDB which are polynomially solvable.

Proposition 3.3. *The fundamental min cut basis problem* MINFDB *in complete graphs G can be solved in $\mathcal{O}(n\,\mathcal{K})$ time, where $\mathcal{K}$ is the complexity of finding a minimum u-v cut.*

Proof. The unconstrained, i.e. non-fundamental, cut basis problem is a relaxation of MINFDB and can by Theorem 2.5 be solved in polynomial time outputing a cut tree T. Since G is a complete graph, T is a spanning tree of G and thus solves MINFDB . □

The second graph type for which MINFDB is polynomially solvable is the *cactus graph* (see Figure 2), i.e., a graph where any pair of different cycles is edge disjoint .

Proposition 3.4. *The fundamental min cut basis problem* MINFDB *in cactus graphs G can be solved in $\mathcal{O}(m \log n)$ time.*

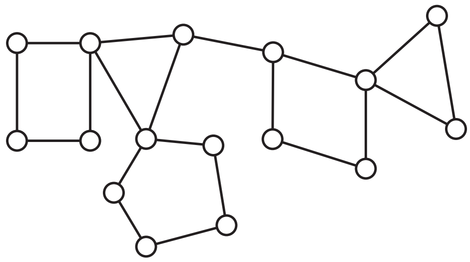

Figure 2: Cactus graph

The proof of this result is an immedate consequence of the Heavy Tree heuristic presented next. This heuristic is based on (3.1) in the proof of Theorem 3.2 which suggests to include edges with large weights w_e in the tree T, since then their weights are only counted once.

Algorithm 3.5. *Heavy Tree Heuristic*
Choose T as maximum weighted tree in the weighted graph (G, w).

Clearly, Algorithm 3.5 solves MinFDB to optimality in cactus graphs (see Figure 2).

If, on the other hand, all weights are equal, then the objective function in edge form (3.1) reduces to

$$\min \sum_{f=(u,v)\in E} l_{uv}^T, \tag{3.2}$$

i.e. we want to minimize the sum of the length of all paths connecting nodes u and v in T with $(u, v) \in E(G)$. As a surrogate for (3.2) we take the objectives of an unweighted min-sum or min-max location problem in the network G.

Algorithm 3.6. *Short Tree Heuristic*

1. *Find the optimal solution $v^* \in V$ of the min-sum location problem* $\min_{v\in V} \sum_{u\in V} d(v, u)$ *or of the min-max location problem* $\min_{v\in V} \max_{u\in V} d(v, u)$.
2. *Choose T as the shortest path tree from v^* to all other nodes $u \in V$.*

Since the min-sum and min-max problem are known in location theory also as median and center problem, we call a tree resulting from Algorithm 3.6 *median short tree* and *center short tree*, respectively. The short tree and the heavy tree heuristics are complementing each other, since one of them is usually producing good solutions if the other does not.

In the last heurstic we start with a cut-tree by solving MinDB as relaxation of MinFDB and apply iteratively edge swaps until the tree becomes a subtree of G. Recall that an *edge swap* $T[e, f]$ consists in removing an edge $e \in E(T)$ and replacing it by an edge $f \in E \setminus (E(T)$ from the fundamental cut of T and e. Obviously, edge swaps maintain the tree property such that the following heuristic provides a feasible solution for MinFDB .

Algorithm 3.7. *Feasible Cut Tree Heuristic*

1. *Find an optimal solution for the unconstrained problem* MinDB *by computing an optimal cut-tree* $T = (V, E(T))$ *using a multi-terminal network flow algorithm (see Section 2).*

2. *If* $E(T) \subseteq E$, *then* T *defines an optimal solution of* MinFDB.

3. *Otherwise, use edge swaps to transform* T *into a subtree of* G.

Besides its application in Algorithm 3.7, edge swaps can be used in metaheuristics such as Local Search and Variable Neighbourhood Search (VNS).
The former checks if a given tree - computed, for instance, by any of the preceding heuristic – is locally optimal. I.e., it computes the total cut weight of all the trees that can be obtained from the given tree through an edge swap. If one of these adjacent trees has a smaller total cut weight, the given cut-tree is replaced by its best neighbour and the search is iterated. The algorithm stops with a locally optimal solution.
In order to escape local optima which are not global ones, we repeat edge swaping a certain number (the *neighbourhood size*) of times without checking on an improvement of the objective value during the process. We then apply local search procedure to the new tree. If a better solution is achieved in this way, we replace the previous locally optimal tree by it. Obviously, the improved quality of the VNS solutions has to be paid for with increased running times.

4 Integer Linear Programming Formulations

In this section, we give three integer programming formulations for the fundamental cut basis problem MinFDB . The first two formulations use the cross product of tree and cut basis formulations. The latter formulations are tied together by a condition which ensures that the the cut basis is, indeed, a fundamental one. In the third formulation we present a possibility to avoid the explicit formulation of tree constraints by using a tree-free characterization of fundamental cut bases.

4.1 Formulation 1

We start this section with an intuitive formulation which uses the following interpretation of the binary variables.

$$x_{ij}^k := \begin{cases} 1 & \text{if edge } (i,j) \in \text{cut } D_k \\ 0 & \text{else} \end{cases}$$

$$y_{ij} := \begin{cases} 1 & \text{if edge } (i,j) \in \text{spanning tree } T \\ 0 & \text{else} \end{cases}$$

$$z_{ik} := \begin{cases} 1 & \text{if node } i \text{ is contained in shore } U_k \text{ of cut } D_k = (U_k, V \setminus U_k) \\ 0 & \text{else} \end{cases}$$

Then problem MinFDB can be formulated as follows.

$$\min \sum_{k=1}^{n-1} \sum_{(i,j)\in E} c_{ij} x_{ij}^k$$

$$\text{such that} \quad \sum_{(i,j)\in E} y_{ij} = n-1 \tag{4.3}$$

$$\sum_{(i,j)\in E(S)} y_{ij} \leq |S|-1 \quad \forall\, S \subset V,\, 2 \leq |S| \leq n \tag{4.4}$$

$$\sum_{k=1}^{n-1} x_{ij}^k \leq 1 + m(1-y_{ij}) \quad \forall (i,j) \in E \tag{4.5}$$

$$\sum_{k=1}^{n-1} x_{ij}^k \geq 1 \quad \forall (i,j) \in E \tag{4.6}$$

$$\sum_{i=1}^{n} z_{ik} \geq 1 \quad \forall k \tag{4.7}$$

$$x_{ij}^k \leq z_{ik} + z_j^k \quad \forall (i,j) \in E, \forall k \tag{4.8}$$

$$x_{ij}^k \leq 2 - z_{ik} - z_j^k \quad \forall (i,j) \in E, \forall k \tag{4.9}$$

$$-x_{ij}^k \leq z_{ik} - z_j^k \quad \forall (i,j) \in E, \forall k \tag{4.10}$$

$$-x_{ij}^k \leq z_j^k - z_i^k \quad \forall (i,j) \in E, \forall k \tag{4.11}$$

$$x_{ij}^k, y_{ij}, z_i^k \in \{0,1\}$$

(4.3) and (4.4) is the well-known rank formulation for spanning trees using the binary variables y_{ij}. The constraints (4.6) to (4.11), and the binary edge-cut variables x_{ij}^k and shore-cut variables z_i^k define a cut base. These sets of constraints define the cross product of characteristic vectors of trees and cuts bases. Constraints (4.5) tie the polytopes together and guarantee that the cut base defined by x and z is, indeed, the fundamental cut base of the tree defined by y.

4.2 ILP Formulation 2

Instead of using the formulations of the tree and cut base polytope of the first formulation any other formulation can be used. The rank formulation is, for instance, from a computational point bad, since we would have to deal with exponentially many constraints. Section 5 contains a discussion on how to deal with this issue.

In the second formulation we follow the same approach as the first one in crossing tree and cut base polytopes. Our goal

We assume in this formulation that the given graph G is biconnected and loopless.

For the spanning tree polytope we use a formulation due to [Grö77]. For $F \subset E$, we denote by G_{ctrF} the graph resulting from G by contraction of F, in which F is deleted and the vertices of $V(F)$ are shrunk to a super node. (Note that if $F \neq E(V(F))$, loops can arise.) Furthermore $\mathcal{H}$ denotes the set of vertex-induced biconnected proper subgraphs of G. Then the spanning tree polytope

$$\mathcal{P}_{ST} \quad = \quad \{y \in \mathbb{R}^m \mid y \text{ satisfies (4.12) - (4.14)}\}$$

given by

$$y(E) = \sum_{e \in E} y_e \quad = \quad n - 1 \tag{4.12}$$

$$y(F) = \sum_{e \in F} y_e \quad \leq \quad |W| - 1 \quad \forall\, H = (W, F) \in \mathcal{H} \text{ s.t. } G_{ctrF} \text{ biconnected} \tag{4.13}$$

$$y_e \quad \geq \quad 0 \quad \forall\, e \in E \text{ s.t. } G \setminus e \text{ biconnected} \tag{4.14}$$

has the dimension $\dim(\mathcal{P}_{ST}) = m - 1$.

A minimal description of the cut polytope $\mathcal{P}_{\mathcal{D}}$ of a simple graph not contractible to K_5 is given as follows (see [BG86]). For a cycle C of G, we call $h \in E \setminus C$ a chord of C if there exist two cycles C_1 and C_2 such that $C_1 \cap C_2 = \{h\}$ and $C_1 \triangle C_2 = C$. We denote by E_3 the set of edges not contained in a triangle, that is, a cycle of size three.
Then $\mathcal{P}_{\mathcal{D}}$ has the form

$$\mathcal{P}_{\mathcal{D}} \quad = \quad \{x \in \mathbb{R}^m \mid x \text{ satisfies (4.15) and (4.16)}\}$$

where

$$\sum_{e \in F} x_e - \sum_{e \in C \setminus F} x_e \quad \leq \quad |F| - 1 \quad \forall \text{ cycles } C \text{ without chord; } \forall\, F \subseteq D,\ |F| \text{ odd} \tag{4.15}$$

$$0 \quad \leq \quad x_e \quad \leq \quad 1 \quad \forall\, e \in E \setminus E_3\,. \tag{4.16}$$

Since there are no parallel edges or loops in G, the cut polytope $\mathcal{P}_{\mathcal{D}}$ is fulldimensional, i.e., $\dim(\mathcal{P}_{\mathcal{D}}) = m$.
Combining the preceding formulations we define

$$\mathcal{Q}_{\mathcal{F}} \quad := \quad \{p = (x, y) = (x^1, \ldots, x^{n-1}, y) \in \mathbb{R}^{nm} \mid p \text{ satisfies (4.17) - (4.24)}\}$$

where

$$y(E) = \sum_{e \in E} y_e \quad = \quad n - 1 \tag{4.17}$$

$$y(F) = \sum_{e \in F} y_e \quad \leq \quad |W| - 1 \quad \forall\, H = (W, F) \in \mathcal{H} \text{ s.t. } G_{ctrF} \text{ biconnected} \tag{4.18}$$

$$\sum_{e \in F} x_e^k - \sum_{e \in C \setminus F} x_e^k \quad \leq \quad |F| - 1 \;\forall \text{ chordless cycles } C, \forall\, F \subseteq C,\, |F| \text{ odd}; \forall\, k \tag{4.19}$$

$$\sum_{e \in E} x_e^k \quad \geq \quad 2 \qquad \forall\, k \tag{4.20}$$

$$\sum_{k=1}^{n-1} x_e^k \quad \geq \quad 1 \qquad \forall\, e \in E \tag{4.21}$$

$$\sum_{k=1}^{n-1} x_e^k \quad \leq \quad 1 + m(1 - y_e) \qquad \forall\, e \tag{4.22}$$

$$0 \quad \leq \quad x_e^k \;\leq\; 1 \qquad \forall\, e \in E \setminus E_3\,;\; \forall\, k \tag{4.23}$$

$$y_e \quad \geq \quad 0 \quad \forall\, e \in E \text{ s.t. } G \setminus e \text{ biconnected} \tag{4.24}$$

Here (4.17) - (4.19) together with (4.23) - (4.24) describe the product $\mathcal{P}_{\mathcal{D}}^{n-1} \times \mathcal{P}_{ST} \subset \mathbb{R}^{nm}$ which contains $\mathcal{P}_{\mathcal{F}}$. New constraints (4.20) - (4.22) are added to link the variables describing the spanning tree to those of the corresponding fundamental cuts to guarantee fundamentality. Note that conditions (4.18) contain in particular the requirement $y_e \leq 1$ for all $e \in E$ such that $G_{ctr\{e\}}$ is biconnected. We have thus the following result.

Proposition 4.1. *The polytope $\mathcal{Q}_{\mathcal{F}}$ is a formulation for $\mathcal{X}_{\mathcal{F}}$, that is, $\mathcal{X}_{\mathcal{F}} = \mathcal{Q}_{\mathcal{F}} \cap \mathbb{Z}^{nm}$.*

In the definition of $\mathcal{Q}_{\mathcal{F}}$, the independence of the $n-1$ cuts does not have to be required explicitly since it is already implied by the fundamentality of the cuts corresponding to the spanning tree T. Note also, that $\mathcal{P}_{\mathcal{D}}^{n-1} \times \mathcal{P}_{ST}$ is – as product of integral polytopes – an integral polytope. But this integrality is destroyed by the additional constraints (4.20-4.22).

We can now easily establish the dimension of polytope $\mathcal{P}_{\mathcal{F}}$.

Theorem 4.2. *Let G be a biconnected, loopless graph. Then the dimension of the fundamental cut basis polytope $\mathcal{P}_{\mathcal{F}}$ is dim $(\mathcal{P}_{\mathcal{F}}) = nm - 1$.*

Proof. The dimension of the product $\mathcal{P}_{\mathcal{D}}^{n-1} \times \mathcal{P}_{ST} \subset \mathbb{R}^{nm}$ is the sum of the dimensions of its components, since the blocks of variables are independent from each other. All vectors in $\mathcal{P}_{\mathcal{F}}$ satisfy equation (4.17), and it hence follows that dim $(\mathcal{P}_{\mathcal{F}}) = \dim(\mathcal{P}_{\mathcal{D}}^{n-1} \times \mathcal{P}_{ST}) = (n-1)m + (m-1) = nm - 1$. □

4.3 ILP Formulation 3

The last formulation is based on a characterization of fundamental cuts which is dual to the one for fundamental cycles given by [HM91], [Sys79] and [Sys81]. Let $\mathcal{D} = \{D_i\}_{i \in I}$ be a cut basis of graph G, where $I = \{1, \dots, n-1\}$.

Lemma 4.3. *Let $\mathcal{D} = \{D_i\}_{i \in I}$ be a cut basis of graph G. Then the following statements are equivalent:*

(i) $\mathcal{D}$ is a fundamental cut basis.

(ii) For every $D_j \in \mathcal{D}$ there exists an $e_j \in D_j$ with $e_j \notin D_i$ for $i \in I \setminus \{j\}$.

Proof. $(i) \Longrightarrow (ii)$: If $\mathcal{D}$ is fundamental then every branch $e \in T$ is contained exclusively in the cut $D(T, e)$ generated by it. Hence every cut of the basis contains at least one edge that does not belong to the other cuts.
$(ii) \Longrightarrow (i)$: Let $H := (V, \cup_{j=1}^{n-1} e_j)$. Then the number of edges of H is $n-1$. Suppose that C is a cycle in H. Then for any $e_k \in C \subset H$ we would get $C \cap D_k = \{e_k\}$, a contradiction to the fact that the number of elements in the intersection of any cycle with any cut is even. It follows that H is a spanning tree that generates the basis $\mathcal{D}$, hence $\mathcal{D}$ is fundamental. □

We use condition (ii) of Lemma 4.3 (which we call in the following the *Syslo condition*) to get an alternative integer programming formulation. Let $M := 2m(n-1)$ and let $\mathcal{X}_{\mathcal{F}_2} \subset \mathbb{B}^M$ be the set of incidence vectors of fundamental cut bases of G, given in the form $\chi = (\chi_{D(T,e_1)}, \dots, \chi_{D(T,e_{n-1})}, \chi_{b_1}, \dots, \chi_{b_{n-1}})$. Here, $\chi_{D(T,e_k)}$ are the incidence vectors of the fundamental cuts $D(T, e_k)$ corresponding to some spanning tree T of G and χ_{b_k} contain the information which edge is the unique branch of T in the fundamental cut $D(T, e_k)$. Let $\mathcal{P}_{\mathcal{F}_2} := conv(\mathcal{X}_{\mathcal{F}_2})$ be its convex hull.

Consider

$$\mathcal{Q}_{\mathcal{F}_2} \quad := \quad \{p = (x,t) = (x^1, \dots, x^{n-1}, t^1, \dots, t^{n-1}) \in \mathbb{R}^M \mid p \text{ satisfies (4.25) - (4.31)}\}$$

where

$$\sum_{e\in F} x_e^k - \sum_{e\in C\setminus F} x_e^k \leq |F|-1 \; \forall \text{ chordless cycles } C, \forall F\subset C, |F| \text{ odd}; \forall k \quad (4.25)$$

$$\sum_{e\in E} x_e^k \geq 2 \qquad \forall k \quad (4.26)$$

$$\sum_{k=1}^{n-1} x_e^k \geq 1 \qquad \forall e\in E \quad (4.27)$$

$$\sum_{e\in E} t_e^k = 1 \qquad \forall k \quad (4.28)$$

$$\sum_{h=1}^{n-1} x_e^h \leq 1+m(1-t_e^k) \qquad \forall e\in E\,;\,\forall k \quad (4.29)$$

$$0 \leq x_e^k \leq 1 \qquad \forall e\in E\,;\,\forall k \quad (4.30)$$

$$0 \leq t_e^k \leq x_e^k \qquad \forall e\in E\,;\,\forall k \quad (4.31)$$

Proposition 4.4. *The polytope $\mathcal{Q}_{\mathcal{F}_2}$ is a formulation for $\mathcal{X}_{\mathcal{F}_2}$, that is, $\mathcal{X}_{\mathcal{F}_2} = \mathcal{Q}_{\mathcal{F}_2} \cap \mathbb{Z}^M$.*

In this formulation we accept an increase in the number of variables compared with formulation 2 in order to eliminate the exponentially many subtour elimination constraints. Constraints (4.28) and (4.29) model the Syslo-condition (ii) of Lemma 4.3 as they make sure that every cut of the basis contains one edge that is not contained in any other of the cuts. Hence the fundamentality of the cuts can be guaranteed without modelling a spanning tree explicitly. But variables t_e^k can be interpreted as to linearize the product $x_e^k y_e$ meaning that "edge e is the branch of cut k". Hence condition (4.28) ensures that every cut contains exactly one branch and by (4.29), it is excluded that a branch belongs to more than one cut.

5 Numerical Results

Similar to the fundamental cycle basis problem MinFCB [ALMM04], exact solutions for the fundamental cut basis problem can only be found for very small instances ($n \leq 7$). We therefore strive to find solution approaches which provide reasonable duality gaps (DG). For this purpose, we use the heuristics

FCT	Feasible Cut Tree
HT	Heavy Tree
MT	Median Tree
CT	Center Tree
VNS	Variable Neighbourhood Search

from Section 3 to provide upper bounds and relaxations

GHT	Gomory Hu Tree (see Section 2)
LPR	LP Relaxation (see Section 4)

as lower bounds. It should be noted that the GHT bound can also be interpreted as a Lagrange bound, since it corresponds to a feasible Lagrange solution with multipliers equal to zero.

We performed the tests on randomly generated graphs with input number of vertices n, edge density p, and weight span $[w_{min}, w_{max}]$. In the following, we will first analyze the lower bounds and then report on our first experiences with duality gaps.

5.1 Computation of lower bounds

In order to compute lower bounds using the LP relaxations of the formulations from Section 4, we have to deal with the exponentially many constraints in these formulation. In Formulations 1 and 2 we replaced the rank constraints (4.3) and (4.4), and (4.12) and (4.13), respectively, by the following set of polynomially many *level constraints* (see, for instance, [GPS00], [Sch05] or [Bun06], where the latter also contains a validity proof).

$$\sum_{e=(u,v)\in E} y_{uv} = n-1 \tag{5.32}$$

$$\text{level}(v) \geq \text{level}(u) + 1 - n + n \cdot y_{uv} \qquad \forall u,v \in V, u \neq v \tag{5.33}$$

$$\sum_{v\in V, v\neq u} y_{uv} = 1 \qquad \forall u \in V, u \neq 1 \tag{5.34}$$

$$\text{level}(u) \in \mathbb{N} \qquad \forall u \in V \tag{5.35}$$

Instead of checking the cycle basis constraints (4.19) (or (4.25)) for all cycles in G we can restrict ourselves to the cycles in an arbitrarily chosen cycle basis of graph G, since the evenness of the intersection inherits itself from a cycle basis to all cycles. Moreover, a cut can be characterized as an inclusionwise minimal set of edges D with the property that $|C \cap D| \neq 1$ for each cycle C. This can be modeled by using conditions (4.19) only for singleton subsets $F \subseteq C$, i.e. sets with $|F| = 1$. We can, therefore, replace the expontially many constraints (4.19) by

$$x_f^k - \sum_{e\in C\setminus\{f\}} x_e^k \leq 0 \;\forall \text{ cycles } C \text{ in a basis}, \forall\, f \in C; \forall\, k \tag{5.36}$$

$$\sum_{e\in C} x_e^k \leq 2, \;\forall \text{ cycles } C \text{ in a basis}; \forall\, k \tag{5.37}$$

If we use for each cycle in the basis characteristic vectors $c^l = (c_e^l)_{e\in E}$ then the preceding constraints can be written as

$$c_f^l x_f^k - \sum_{e \in E \setminus \{f\}} c_e^l x_e^k \leq 0 \; \forall f \in E; \forall k; \forall l \tag{5.38}$$

$$\sum_{e \in E} c_e^l x_e^k \leq 2, \; \forall k; \forall l. \tag{5.39}$$

Table 1 shows some results comparing the resulting LP relaxation bounds (LPR) with the Gomory/Hu lower bounds (GHT). In this table – and in all our computations – GHT is the clear winner.

	Average Degree	GHT/LPR
n=30, p=13.8%	4	1.68
n=30, p=27.6%	8	1.88
n=30, p=41.4%	12	1.89
n=30, p=55.2%	16	1.90
n=30, p=69.0%	20	1.91
n=30, p=82.8%	24	1.91
n=30, p=96.6%	28	1.92
n=30, p=100.0%	29	1.92
n=50, p=6.1%	3	1.78
n=50, p=8.2%	4	1.92

Table 1: Quality of the lower bounds measured by the ration GHT/LPR (in all instances $w \in [1, 10]$)

That the GHT bound would do well for (almost) complete graphs is in light of Proposition 3.3 no surprise, since the solution is very likely to be optimal not only for the relaxation MinDB but also for MinFDB . But even for sparse graphs the Gomory Hu cut tree yield much better bounds than the LP relaxations. The ration between the bounds gets also larger for increasing number n of vertices.

One reason for the bad performance of LP relaxation is the fact that the inequality $\sum_{k=1}^{n-1} x_{ijk} \geq 1 \quad \forall (i,j) \in E$ (4.5) is satisfied with equality for all tested instances. I.e. in the integer case every edge has to be contained in at least one cut, every chord is contained in at least two cuts, whereas in the relaxed case an edge is contained in a cut with "just a percentage" and these add up to 1. This way we considerably underestimate the chords.

It takes much time to solve the LP relaxation, which is not surprising, since for a graph with 100 vertices and an edge density of 0.5 we get, for instance, over 1.5 million functional constraints and about 750000 binary variables.

As a consequence of the lower bound comparisons, the gap computations which are discussed next are only done using the GHT lower bound.

5.2 Duality gap

In the test 20 random graphs were generated for each combination of parameters. The figures shown in the tables are the mean values of the corresponding duality gaps.

	DG Heuristic in %				DG VNS in %			
n	FCT	HT	MT	CT	FCT	HT	MT	CT
5	0.3	4.9	9.1	9.4	0.1	0.1	0.1	0.1
10	4.2	37.5	2.1	4.3	2.6	2.1	1.7	1.7
15	10.1	74.0	8.0	16.5	7.3	11.7	6.5	11.3
20	8.8	97.8	5.8	15.8	7.1	13.1	5.2	12.7
25	16.1	232.3	09.8	20.5	12.3	13.9	9.1	18.4
30	12.1	250.5	9.6	19.4	10.7	13.3	9.0	17.5

Table 2: Influence of the graph size ($p = 0.75$, w in [1,10], $k = 5$)

	DG Heuristic in %	
n	FCT	MT
40	12.7	10.0
60	16.9	10.5
80	19.0	12.6
100	18.8	13.5

Table 3: Influence of the graph size ($p = 0.75$, w in [1,10], $k = 0$)

Tables 2 and 3 demonstrate the influence of the graph size on the performance of the heuristics. The larger the graph the worse the behaviour of the heuristics which becomes especially apparent for the heavy tree heuristic. This is due to the increasing importance of having short paths which is not accounted for in the heavy tree. In the worst case the maximum spanning tree is a Hamiltonian path.
The median tree yields the best results for the heuristic and the variable neighbourhood search except for instances with five vertices where the feasible cut tree and the heavy tree are preferable. For larger graphs we tested the most promising heuristics, Feasible Cut Tree and Median Tree and neglected VNS due to computation times, see Table 3.
The influence of the graph density is shown in Tables 4 and 5. The feasible cut tree yields good solutions, in complete graphs due to Proposition 3.3 obvioiusly the optimal one. The initial short trees (Median and Center) become better, the heavy tree worse with increasing density. Furthermore, we can observe that the heavy tree yields better solutions than the short trees in graphs with only a few vertices.

The weight intervals tested in Table 6 do not influence the performance of the different heuristics. The best solutions are obtained by the median tree, the second best solutions

p	DG Heuristic in % FCT	HT	MT	CT	DG VNS in % FCT	HT	MT	CT
0.5	0.0	0.9	11.2	11.1	0.0	0.0	0.0	0.0
0.75	0.3	4.9	9.1	9.4	0.1	0.1	0.1	0.1
1.0	0.0	5.9	9.6	9.6	0.0	0.0	0.1	0.2

Table 4: Influence of the density in ($n = 5$, w in [1,10], $k = 5$, $k = 0$ for $p = 0.5$)

p	DG Heuristic in % FCT	HT	MT	CT	DG VNS in % FCT	HT	MT	CT
0.25	16.0	14.1	27.4	34.4	4.4	4.6	4.5	5.0
0.5	19.6	50.7	17.3	25.6	10.9	10.3	9.9	12.4
0.75	10.1	74.0	8.0	16.5	7.3	11.7	6.5	11.3
1.0	0.0	76.2	1.4	1.4	0.0	0.8	1.4	1.4

Table 5: Influence of the density in ($n = 15$, w in [1,10], $k = 5$)

by the feasible cut tree. The heavy tree solution does not yield an acceptable duality gap.

For the tests shown in Table 7 an edge weight is within the first interval with probability 75% and within the second interval with 25%. With increasing distance of the intervals the short trees are less and the heavy tree is more efficient. Heavy tree and feasible cut tree show the best behaviour if improved by variable neighbourhood search.

The last investigation concerning the influence of the weight is shown in Table 8. The probability of the weight to be in the first of two intervals has been modified. Except for the cases with probability 0 and 100%, the heavy tree behaves better and the short trees behave worse for increasing probability of a lower weight. The performance rate between heavy and short tree has a break even at probability 75% .

Enlarging the size of the neighbourhood to be combed improves the VNS-duality gaps (see Table 9) but has to be paid off by higher computation times.

Summing up the results of our numerical tests, the feasible cut tree is the most reliable heuristic. The center tree is never better than the median tree. The median tree yields good results except for very small and sparse graphs and spread weight intervals. In these cases the heavy tree is more efficient. Overall, the worst case solutions are much better of the median than of the heavy tree. The VNS-duality gap of the respectively best procedure is always below 10% with a neighbourhood size of only five. For larger graphs the initial solution of median tree still provides a duality gap of 13.5% without VNS.

w	DG Heuristic in % FCT	HT	MT	CT	DG VNS in % FCT	HT	MT	CT
$[1, 10]$	10.1	74.0	8.0	16.5	7.3	11.7	6.5	11.3
$[1, 100]$	11.9	70.2	9.4	16.7	8.0	11.8	7.2	11.0
$[1, 1000]$	11.0	63.8	7.6	16.1	7.5	10.8	6.2	11.4
$[1, 10000]$	9.9	74.2	7.2	13.9	7.0	9.4	5.7	9.9

Table 6: Influence of the weight ($n = 15$, $p = 0.75$, $k = 5$)

w	DG Heuristic in % FCT	HT	MT	CT	DG VNS in % FCT	HT	MT	CT
$[1, 10] \cup [91, 100]$	15.2	28.8	14.4	26.1	3.9	4.6	5.7	7.5
$[1, 10] \cup [991, 1000]$	13.1	26.4	24.5	29.0	2.6	2.3	7.6	6.9
$[1, 10] \cup [9991, 10000]$	8.3	18.3	31.9	48.9	2.3	2.4	9.6	10.3

Table 7: Influence of the weight ($n = 15$, $p = 0.75$, $k = 5$)

6 Conclusion and Further Research

In this paper we considered the unconstrained and the fundamental cut basis problems in undirected graphs. The unconstrained problem is solved by applying a multiterminal network flow algorithm and using the cuts defined by the resulting minimum cut tree. The fundamental cut basis problem was shown to be NP-hard. Two polynomially solvable special cases are the complete graphs and cactus graphs. Several heuristics and integer programming formulatios were proposed. The resulting lower and upper bounds were used in numerical test which showed that a duality gap of less than 10% can be achieved combining the proposed heuristics with variable neighbourhood search.

The numerical tests show that the Gomory/Hu lower bounds obtained by the relaxation of the fundmentality outperforms the lower bound obtained by linear programming relaxation. Since the Gomory/Hu bounds are special cases of Lagrangian bounds (with Langrangian multipliers all equal to zero), this indicates the solution of the Lagranian dual as a promissing research area.

Another interesting area is the generalization of the presented formulations for fundamental cut bases to fundamental circuit bases in binary matroids. The reader is refered to [Bun06] for more details in this matter.

References

[ALMM04] E. Amaldi, L. Liberti, F. Maffioli, and N. Maculan. Algorithms for the minimum fundamental cycle basis problem. CTW2004, Menaggio, June 2004 (full

p_1	DG Heuristic in % FCT	HT	MT	CT	DG VNS in % FCT	HT	MT	CT
0.0	7.7	84.0	8.0	16.6	7.0	13.3	7.4	13.3
0.45	11.5	75.4	7.3	15.4	6.3	8.6	6.0	10.5
0.5	6.1	65.4	5.0	8.1	4.1	6.4	4.5	5.9
0.55	12.1	60.8	7.8	10.0	5.1	6.3	5.3	8.5
0.6	17.2	43.3	13.8	21.7	7.0	5.8	8.7	9.9
0.65	10.2	42.1	11.5	16.9	3.3	5.0	6.2	7.1
0.7	7.9	36.3	10.4	14.4	2.5	4.3	5.4	5.2
0.75	13.1	26.4	24.5	29.0	2.6	2.3	7.6	6.9
0.8	13.9	10.3	37.2	39.5	1.0	0.7	8.0	5.5
0.85	7.2	4.8	60.9	71.9	0.5	0.5	5.4	3.2
0.9	1.5	3.0	65.5	76.1	0.2	0.4	2.7	1.6
0.95	2.4	10.6	63.3	74.6	1.2	1.4	1.3	1.9
1.0	10.1	74.0	8.0	16.5	7.3	11.7	6.5	11.3

Table 8: Influence of the weight spread ($n = 15$, $p = 0.75$, w in $[1, 10] \cup [991, 1000]$, $k = 5$)

k	DG Heuristic in % FCT	HT	MT	CT	DG VNS in % FCT	HT	MT	CT
0	9.1	63.7	7.3	15.7	6.5	8.5	6.2	12.8
5	10.1	74.0	8.0	16.5	7.3	11.7	6.5	11.3
10	11.1	68.2	9.3	18.9	7.6	9.4	7.3	9.9
15	8.1	70.3	4.5	10.4	4.8	6.5	3.7	4.5

Table 9: Influence of the neighbourhood size in ($n = 15$, $p = 0.75$, w in [1,10])

paper submitted to Discrete Applied Mathematics), 2004.

[BG86] F. Barahona and M. Grötschel. On the cycle polytope of a binary matroid. *Journal of Combinatorial Theory B*, 40:40–62, 1986.

[Bun06] F. Bunke. *Circuit bases problems in binary matroids.* Ph.D. Thesis, Department of Mathematics, University of Kaiserslautern, 2006.

[DKP95] K.N. Deo, N. Kumar, and J. Parson. Minimum-length fundamental-cycle set problem: New heuristics and an empirical investigation. *Congressus Numerantium*, 107:141–154, 1995.

[DPK82] M.P.N. Deo, G.M. Prabhu, and M.S. Krishnamoorty. Algorithms for generating fundamental cycles in a graph. *ACM Transactions on Mathematical Software*, 8(1):26–42, 1982.

[GA03] G. Galbiati and E. Amaldi. On the approximability of the minimum fundamental cycle basis problem. In *Proceedings of the Workshop on Approximation and Online Algorithms WAOA*, volume 2909 of *Lecture Notes in Computer Science*. Springer, 2003.

[Gal01] Giulia Galbiati. On min-max cycle bases. In P. Eades and T. Takaoka, editors, *Proceedings of ISAAC*, volume 2223 of *Lecture Notes in Computer Science*, pages 116–123. Springer, 2001.

[GH61] R.E. Gomory and T.C. Hu. Multi-terminal network flows. *J. Soc. Indust. Appl. Math.*, 9(4):551–570, 1961.

[GH02] A. Golynski and J.D. Horton. A polynomial time algorithm to find the minimum cycle basis of a regular matroid. In *Algorithm Theory - SWAT 2002 : 8th Scandinavian Workshop on Algorithm Theory, Turku, Finland, July 3-5, 2002. Proceedings*, volume 2368 / 2002 of *Lecture Notes in Computer Science*, pages 200–209. Springer, Heidelberg, 2002.

[GPS00] C. Gueret, C. Prins, and M. Sevaux. *Applications of optimization with Xpress-MP*. Dash Optimization Ltd, 2000.

[GPS02] A.M.H. Gerards, J.C. De Pina, and A. Schrijver. Shortest circuit bases of graphs. Manuscript, 2002.

[Grö77] M. Grötschel. *Polyedrische Charakterisierungen kombinatorischer Optimierungsprobleme*, volume 36 of *Mathematical Systems in Economics*. Anton Hain, 1977.

[Gus90] D. Gusfield. Very simple methods for all pairs network flow analysis. *SIAM Journal on Computing*, 19:143–155, 1990.

[HM91] D. Hartvigsen and R. Mardon. Cycle bases from orderings and coverings. *Discr. Math.*, 94:81–94, 1991. (Quelle: MAT Z 1124; $\surd$).

[Hor87] J.D. Horton. A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM Journal on Computing*, 16(2):358–366, 1987.

[HQ85] H.W. Hamacher and M. Queyranne. K best solutions to combinatorial optimization problems. *Annals of Operations Research*, (4):123–143, 1985.

[KMMP04] T. Kavitha, K. Mehlhorn, D. Michail, and K. E. Paluch. A faster algorithm for minimum cycle basis of graphs. In J. Diaz, J. Karhumäki, A. Lepistö, and D. Sanella, editors, *ICALP, Lecture Notes in Computer Science*, volume 3142, pages 846–857. Springer, 2004.

[Sch05] A.M. Schwahn. Minimum fundamental cut basis problem. Diploma Thesis, Technische Universität Kaiserslautern, 2005.

[Sys79] M.M. Sysło. On cycle bases of a graph. *Networks*, 9:123–132, 1979.

[Sys81] M.M. Sysło. On some problems related to fundamental cycle sets of a graph. In *Chartrand, R., Theory of Applications of Graphs: Fourth International Graph Theory Conference, Proceedings*, pages 577–588. Wiley, 1981.