A Comparative Study of Linear and Semidefinite Branch-and-Cut Methods for Solving the Minimum Graph Bisection Problem

Michael Armbruster[1], Marzena Fügenschuh[2], Christoph Helmberg[1], and Alexander Martin[2]

[1] Chemnitz University of Technology
Department of Mathematics, D-09107 Chemnitz, Germany
helmberg@mathematik.tu-chemnitz.de
[2] Technische Universität Darmstadt
Department of Mathematics, D-64289 Darmstadt, Germany
martin@mathematik.tu-darmstadt.de

Abstract. Semidefinite relaxations are known to deliver good approximations for combinatorial optimization problems like graph bisection. Using the spectral bundle method it is possible to exploit structural properties of the underlying problem and to apply, even to sparse large scale instances, cutting plane methods, probably the most successful technique in linear programming. We set up a common branch-and-cut framework for linear and semidefinite relaxations of the *minimum graph bisection problem*. It incorporates separation algorithms for valid inequalities presented in the recent study [**?**] of the facial structure of the associated polytope. Extensive numerical experiments show that the semidefinite branch-and-cut approach outperforms the classical simplex approach on a clear majority of the sparse large scale test instances. On instances from compiler design the simplex approach is faster.
Keywords: branch and cut algorithms, cutting plane algorithms, polyhedral combinatorics, semidefinite programs

1 Introduction

Let $G = (V, E)$ be an undirected graph with $V = \{1, \dots, n\}$ and $E \subseteq \{\{i, j\} : i, j \in V, i < j\}$. For given vertex weights $f_v \in \mathbb{N}_\cup\{0\}$, $v \in V$, and edge costs $w_{\{i,j\}} \in \mathbb{R}$, $\{i, j\} \in E$, a partition of the vertex set V into two disjoint clusters S and $V \setminus S$ with sizes $f(S) \leq F$ and $f(V \setminus S) \leq F$, where $F \in \mathbb{N} \cap \left[\frac{1}{2} f(V), f(V)\right]$, is called a *bisection*. Finding a bisection such that the total cost of edges in the cut $\delta(S) := \{\{i, j\} \in E : i \in S \wedge j \in V \setminus S\}$ is minimal is the *minimum bisection problem* (MB). The problem is known to be NP-hard [**?**]. The polytope associated with MB,

$$P_{\mathrm{B}} := \operatorname{conv}\left\{ y \in \mathbb{R}^{|E|} : y = \chi^{\delta(S)},\, S \subseteq V,\, f(S) \leq F,\, f(V \setminus S) \leq F \right\},$$

where $\chi^{\delta(S)}$ is the incidence vector of the cut $\delta(S)$ with respect to the edge set E, is called the *bisection cut polytope*. MB as well as P_{B} are related to other problems and polytopes already known in the literature. Obviously, the bisection cut polytope is contained in the *cut polytope* [?,?]

$$P_{\mathrm{C}} := \operatorname{conv}\left\{y \in \mathbb{R}^{|E|} : y = \chi^{\delta(S)}, S \subseteq V\right\}.$$

If $F = f(V)$ then MB is equivalent to the maximum cut problem (using the negative cost function) and $P_{\mathrm{B}} = P_{\mathrm{C}}$. For $F = \lceil \frac{1}{2} f(V) \rceil$ MB is equivalent to the *equipartition problem* [?] and the bisection cut polytope equals the *equipartition polytope* [?,?]. Furthermore, MB is a special case of the minimum node capacitated graph partitioning problem (MNCGP) [?], where more than two clusters are available for the partition of the node set and each cluster has a common limited capacity. The objective in MNCGP is the same as in MB, i.e., to minimize the total cost of edges having endpoints in distinct clusters. Finally, we mention the *knapsack polytope* [?]

$$P_{\mathrm{K}} := \operatorname{conv}\left\{x \in \{0,1\}^{|V|} : \sum_{v \in V} f_v x_v \leq F\right\},$$

which plays a fundamental role in valid inequalities for P_{B}. Graph partitioning problems in general have numerous applications, for instance in numerics [?], VLSI-design [?], compiler-design [?] and frequency assignment [?]. A large variety of valid inequalities for the polytope associated with MNCGP is known [?,?,?,?] and, since MB is a special case of MNCGP, all those inequalities are also valid for P_{B}. A recent successful study of a combined semidefinite polyhedral branch-and-cut approach for max-cut is [?], it is designed for rather dense graphs with up to 400 nodes. In contrast, our semidefinite branch-and-cut approach is applicable to sparse graphs with up to 2000 nodes. In addition, we present a direct comparison with an LP approach within the same branch-and-cut environment where both approaches use the same separation routines.

In [?] we give a detailed analysis of P_{B} including several classes of new and facet-defining inequalities. We summarize these results and those from the literature in Sect. ??. We use these inequalities to derive and strengthen two relaxations for MB. One is based on an integer programming, the second on a semidefinite programming formulation. We develop in Sect. ?? both an LP-based branch-and-cut algorithm and an SDP based branch-and-cut algorithm using the same framework SCIP [?]. In Sect. ?? we give a comprehensive computational comparison of both approaches on various test instances with some surprising outcomes.

2 Valid Inequalities for P_{B}

A large variety of valid inequalities for the cut polytope, the equipartition polytope, and the polytope associated with MNCGP is known: cycle inequalities [?] of the cut polytope; tree, star, and cycle inequalities [?] as well as suspended tree and path block cycle inequalities [?,?] for the equipartition polytope; tree, star, cycle with ear, cycle with tails, and knapsack tree inequalities [?] valid for the polytope associated with MNCGP. Since MB is a special case of MNCGP and $P_{\mathrm{B}} \subseteq P_{\mathrm{C}}$ the bisection cut polytope inherits most of the valid inequalities listed above.

For convenience we cite the cycle inequalities which we will later use in both models for MB. Let the subgraph $C = (V_C, E_C)$ be a cycle in G. Let D be a subset of E_C such that $|D|$ is odd. Then the *cycle inequality*

$$\sum_{e \in D} y_e - \sum_{e \in E_C \setminus D} y_e \leq |D| - 1 \tag{1}$$

is a valid inequality for the cut polytope P_{C}.

The cut structure implies that whenever there is a walk between two nodes of the graph with an even number of edges in the cut, the two end-nodes of the walk have to be in the same cluster. In particular, given a special root node r, a walk P_{rv} in G to some node v, an edge subset $H_v \subseteq P_{rv}$ of even cardinality, and an incidence vector y of a cut; if the term

$$1 - \sum_{e \in P_{rv} \setminus H_v} y_e - \sum_{e \in H_v} (1 - y_e) \tag{2}$$

evaluates to one then r and v are on the same side of the cut; for nodes in opposite clusters, it is at most zero. In [?] this is used to set up an inequality linking the cut structure and the capacity constraint on the node weights.

Proposition 1 (bisection knapsack walk inequality [?]). *Let $\sum_{v \in V} a_v x_v \leq a_0$ be a valid inequality for the knapsack polytope P_{K} with $a_v \geq 0$ for all $v \in V$. For a subset $V' \subseteq V$, a fixed root node $r \in V'$, walks $P_{rv} \subseteq E$, and sets $H_v \subseteq P_{rv}$ with $|H_v|$ even, the* bisection knapsack walk inequality

$$\sum_{v \in V'} a_v \left(1 - \sum_{e \in P_{rv} \setminus H_v} y_e - \sum_{e \in H_v} (1 - y_e) \right) \leq a_0 \ . \tag{3}$$

is valid for the polytope P_{B}.

Given a root node r and a vector $y \in [0,1]^{|E|}$ the optimal walks P_{rv} and subsets H_v maximizing (**??**) can be found in polynomial time with an algorithm that follows the one for separating cycle inequalities [?].

The *knapsack tree inequalities* of [?] form a special case, where the walks P_{rv} are taken from a tree (T, E_T) of G rooted at r and $H_v = \emptyset$ for all $v \in V'$,

$$\sum_{v \in T} a_v \left(1 - \sum_{e \in P_{rv}} y_e \right) \leq a_0 \ . \tag{4}$$

Following [?], one may trivially strengthen the coefficients of (**??**) to

$$\sum_{e \in E_T} \min \left\{ \sum_{v : e \in P_{rv}} a_v, \sum_{v \in T} a_v - a_0 \right\} y_e \geq \sum_{v \in T} a_v - a_0, \tag{5}$$

we call this a *truncated knapsack tree inequality*. A less obvious strengthening exploits the dependence of the coefficients in (**??**) on the choice of the root node, which we express by the notation

$$\alpha_0 := \sum_{v\in T} a_v - a_0, \qquad \alpha_e^r := \min\{\sum_{v: e\in P_{rv}} a_v,\, \alpha_0\},\, e\in E_T, \tag{6}$$

The strongest form is achieved if r enforces a sort of balance with respect to the cumulated node weights on the paths to r.

Theorem 2. [?] *Let (T, E_T) be a tree in G. The strongest truncated knapsack tree inequality, with respect to the knapsack inequality $\sum_{v\in V} a_v x_v \le a_0$, defined on (T, E_T) is obtained for a root $r \in \mathcal{R} := \mathrm{Argmin}_{v\in T} \sum_{e\in E_T} \alpha_e^v$. That is, if $r\in\mathcal{R}$ then $\sum_{e\in E_T} \alpha_e^s y_e \ge \sum_{e\in E_T} \alpha_e^r y_e \ge \alpha_0$ holds for all $s\in T$ and all $y\in P_{\mathrm{B}}$. In particular, $\sum_{e\in E_T} \alpha_e^r y_e = \sum_{e\in E_T} \alpha_e^s y_e$ holds for all $r, s\in\mathcal{R}$ and all $y\in P_{\mathrm{B}}$.*

The elements of the set $\mathcal{R}$ are called *minimal roots* of a given tree (T, E_T), and by Theorem **??** all minimal roots of (T, E_T) deliver the same strongest truncated knapsack tree inequality. Additional structural results allow to locate minimal roots algorithmically at almost no cost. This strengthening proved highly effective in our experiments. Note, if the inequality induces a facet then r is a minimal root by Theorem **??**. In some cases the minimal root condition is also sufficient. In order to state this result, call a path in (T, E_T) *branch-less*, if its inner nodes are all of degree 2 in the tree.

Theorem 3. [?] *Assume that $G = (T, E_T)$ is a tree rooted at a node $r\in T$, $f_v = 1$ for all $v\in T$ and $\frac{|T|}{2} + 1 \le F < |T|$. The truncated knapsack tree inequality $\sum_{e\in E} \min\{|\{v : e\in P_{rv}\}|, |V| - F\} \ge |V| - F$ is facet-defining for P_{B} if and only if one of the following conditions is satisfied:*

(*a*) *r is a minimal root and (T, E_T) satisfies the so-called* branch-less path condition*: each branch-less path with F nodes has one end-edge that is a leaf in (T, E_T),*
(*b*) *$F = |T| - 1$.*

To motivate a strengthening for general bisection knapsack walk inequalities consider the case of a disconnected graph with two components, one of them being a single edge $\{u, v\}$, the other connected one being $V' = V\setminus\{u, v\}$. If $y_{uv} = 1$ then u and v belong to different clusters and therefore the capacity remaining for the clusters in V' (e.g. the right-hand side of (**??**)) can be reduced to $F - \min\{f_u, f_v\}\, y_{uv}$. To generalize this idea we define for $\bar{G}\subseteq G$ with $\bar{V}\subseteq V$, $\bar{E}\subseteq E(\bar{V})$ and $a\in \mathbb{R}_+^{|\bar{V}|}$ a function $\beta_{\bar{G}} : \{0,1\}^{|\bar{E}|} \to \mathbb{R}\cup\infty$ with

$$\beta_{\bar{G}}(y) = \inf\left\{a(S), a(\bar{V}\setminus S) : S\subseteq\bar{V}, \max\left\{a(S), a(\bar{V}\setminus S)\right\} \le a_0, y = \chi^{\delta_{\bar{G}}(S)}\right\}.$$

Now we look at the convex envelope $\breve{\beta}_{\bar{G}} : \mathbb{R}^{|\bar{E}|} \to \mathbb{R}\cup\infty$ of $\beta_{\bar{G}}(y)$, i.e.,

$$\breve{\beta}_{\bar{G}}(y) = \sup\left\{\breve{\beta}(y) :\ \breve{\beta} : \mathbb{R}^{|\bar{E}|} \to \mathbb{R},\ \breve{\beta} \text{ convex},\ \breve{\beta}(z) \le \beta_{\bar{G}}(z) \text{ for } z\in\{0,1\}^{|\bar{E}|}\right\}\ .$$

Note that $\breve{\beta}_{\bar{G}}$ is a piecewise linear function on its domain.

Proposition 4 (capacity reduced bisection knapsack walk inequality [?]). *Let* $\sum_{v\in V} a_v x_v \leq a_0$ *with* $a_v \geq 0$ *for all* $v \in V$ *be a valid inequality for the knapsack polytope* P_{K}*. Let* V_0 *be a non-empty subset of* V *and* $r \in V_0$*. Select subgraphs* $(V_l, E_l) = G_l \subset G$ *with pairwise disjoint sets* V_l*,* $V_l \cap V_0 = \emptyset$ *and* $E_l \subseteq E(V_l)$ *for* $l = 1, \ldots, L$*. Find for each* l *an affine minorant for the convex envelope* $\check{\beta}_{G_l}$ *such that*

$$c_0^l + \sum_{e\in E_l} c_e y_e \leq \check{\beta}_{G_l}(y) \tag{7}$$

holds for all y *in* P_{B}*. Then the* capacity reduced bisection knapsack walk inequality

$$\sum_{v\in V_0} a_v \left(1 - \sum_{e\in P_{rv}\setminus H_v} y_e - \sum_{e\in P_{rv}\cap H_v} (1-y_e)\right) \leq a_0 - \sum_{l=1}^{L} (c_0^l + \sum_{e\in E_l} c_e y_e) \tag{8}$$

is valid for P_{B}*.*

In certain cases it is possible to establish a full description of $\check{\beta}_{\bar{G}}$ via a complete description of the cluster weight polytope defined as follows. Given a graph $G = (V, E)$ with non-negative node weights $a_v \in \mathbb{R}$ for all $v \in V$. For a set $S \subseteq V$ we define $h(S) := (a(S), (\chi^{\delta(S)})^T)^T \in \mathbb{R}^{|E|+1}$. With respect to a given $a_0 \in \mathbb{R}$ we call

$$P_{\mathrm{CW}} = \mathrm{conv}\,\{\, h(S) : \, S \subseteq V,\, a(S) \leq a_0,\, a(V\setminus S) \leq a_0 \,\}$$

the *cluster weight polytope*.

In [?], a full description of $P_{\mathrm{CW}}(\bar{G})$ is given for the special case that the subgraph $\bar{G} = (\bar{V}, \bar{E})$ is a star centered at some node $r \in \bar{V}$, $a \geq 0$, and a_0 satisfies $a(\bar{V}) \leq a_0$. In order to state the nontrivial facets of this case, assume $a(\bar{V}\setminus\{r\}) > a_r$ and call a triple $(V_p, \bar{v}, V_n)$ feasible if it fulfills $\bar{V} = \{r, \bar{v}\} \,\dot{\cup}\, V_p \,\dot{\cup}\, V_n$ and $a(V_p) \leq \frac{1}{2} a(\bar{V}) < a(V_p) + a_{\bar{v}}$. For all feasible triples $(V_p, \bar{v}, V_n)$ the inequalities

$$y_0 - \sum_{v\in V_p} a_v y_{rv} - \big(a(\bar{V}) - 2a(V_p) - a_{\bar{v}}\big)\, y_{r\bar{v}} + \sum_{v\in V_n} a_v y_{rv} \geq 0 \tag{9}$$

are the facet-inducing inequalities for $P_{\mathrm{CW}}(\bar{G})$. Thus, inequalities (**??**) form the best linear minorants (**??**) to be used in **??** in this case.

In our experiments this rather involved strengthening technique proved far less effective than the simple root strengthening for knapsack tree inequalities, but this may be due to the dominating non-negative cost structure in our experiments.

3 Linear and Semidefinite Relaxations for MB

The linear relaxation for MB is derived from the following integer linear programming formulation. We select a node $s \in V$ and extend E so that s is adjacent to all other nodes in V, setting the weights $w(\cdot)$ of new edges to zero. We introduce binary variables y_{ij} for all $ij \in E$ and require that $y_{ij} = 1$ if nodes i and j are in different clusters and $y_{ij} = 0$ otherwise. The capacity constraints on the two clusters can then be formulated as

$$f_s + \sum_{i \in V \setminus \{s\}} f_i(1 - y_{is}) \leq F \,, \tag{10}$$

$$\sum_{i \in V \setminus \{s\}} f_i y_{is} \leq F \,. \tag{11}$$

Thus we obtain the following integer linear model for MB.

$$\begin{aligned} \min \quad & \sum_{e \in E} w_e y_e \\ \text{s.t. } & (\textbf{??}), (\textbf{??}), (\textbf{??}), \\ & y \in \{0,1\}^{|E|} \,. \end{aligned} \tag{12}$$

The cycle inequalities (**??**) make sure that each solution to (**??**) corresponds to an incidence vector of a cut in G. (**??**) and (**??**) require that this cut is a bisection cut. Although the number of all valid cycle inequalities for P_{B} is exponential in $|E|$, the inequalities can be separated in polynomial time [**?**].

Our semidefinite relaxation for MB follows the classical approach of [**?**]. Given the weighted adjacency matrix W of G, represent a bipartition by $x \in \{-1,1\}^n$ with $x_i = -1$ if $x \in S$ and $x_i = 1$ if $x \in V \setminus S$. Then an integer quadratic model for MB reads

$$\min \left\{ \sum_{i<j} w_{ij} \frac{1 - x_i x_j}{2} : \left| f^T x \right| \leq 2F - f(V), x \in \{-1,1\}^{|V|} \right\}. \tag{13}$$

Rewrite $\sum_{i<j} w_{ij} \frac{1-x_i x_j}{2}$ as $\left\langle \frac{1}{4} L, x x^T \right\rangle$, where $L = \mathrm{Diag}(We) - W$ is the weighted Laplace matrix of G, and relax $x x^T$ for $x \in \{-1,1\}^n$ to $X \succeq 0$ with $\mathrm{diag}(X) = e$ to obtain

$$\begin{aligned} \min \ & \left\langle \tfrac{1}{4} L, X \right\rangle \\ \text{s.t. } & \left\langle f f^T, X \right\rangle \leq (2F - f(V))^2, \\ & \mathrm{diag}(X) = e \,, \\ & X \succeq 0 \,. \end{aligned} \tag{14}$$

The framework employs the same separation algorithms for (**??**) and (**??**) by transforming a $\bar{X}$ to a $\bar{y}$ via $\bar{y}_{ij} = \frac{1 - \bar{X}_{ij}}{2}$ for all $\{i,j\} \in E$. Separated inequalities $\sum_{\{i,j\} \in E} k_{ij} y_{ij} \leq k_l$ are translated into constraints $\langle K, X \rangle \leq k_s$ for the primal semidefinite relaxation by $K_{ij} = -\frac{1}{2} k_{ij}$ for all $\{i,j\} \in E$ and $k_s = 2k_l - \sum_{\{i,j\} \in E} k_{ij}$.

Our branch-and-cut implementation using the linear relaxation follows basically standard techniques known in the community. Our implementation with the semidefinite relaxation is, however, not straightforward and involves many details of which we sketch a few. In contrast to [?], where a standard polyhedral bundle method is used together with a rather expensive semidefinite oracle, we solve the semidefinite relaxation approximately by applying the spectral bundle method [?,?] to the dual of (??) in its equivalent form as an eigenvalue optimization problem,

$$-\min_{\substack{z\in\mathbb{R}^{|V|}\\ p\geq 0}} |V|\lambda_{\max}\left(-\frac{1}{4}L+\mathrm{Diag}(z)-ff^Tp\right)-\langle e,z\rangle+(2F-f(V))^2\,p\ . \quad (15)$$

In this setting, the oracle is a Lanczos method for computing extremal eigenvalues and corresponding eigenvectors of large structured matrices; we use the eigenvectors to form a semidefinite cutting model. Any dual feasible solution of (??) yields a valid lower bound for MB. The decisive step in the use of the spectral bundle method in branch-and-cut is to exploit its restarting properties and its primal approximate solution.

While solving (??) the bundle method aggregates the eigenvector information to an approximate primal solution $\tilde{X}$ of (??) of the form

$$\tilde{X}=PUP^T+\alpha\overline{W}\succeq 0\ , \quad (16)$$

where $P\in\mathbb{R}^{n\times k}$, $P^TP=I$, holds a basis of the aggregated eigenvectors, $U\in S_+^k$, $\alpha\geq 0$ so that $\mathrm{tr}\,U+\alpha=|V|$, and $\overline{W}\in S_+^n$ is sparse with $\mathrm{tr}\,\overline{W}=1$. The software allows to choose the support of $\overline{W}$. We start with the support of L and extend it on the fly by further off-diagonal elements (edges) that promise to be useful in the separation of cycle inequalities; this proved to be highly effective already in [?]. The approximate solution $\tilde{X}$ will in general satisfy the constraints approximately only. On the one hand this may result in off-diagonal elements $\tilde{X}_{ij}$ outside of the interval $[-1,1]$, so $\tilde{X}$ has to be rounded or truncated before the standard separation routines can be applied (see above for the transformation to y). On the other hand, a separated inequality may still be violated after the next optimization run, so precautions have to be taken against separating the same inequality again and again. Such aspects have been addressed in [?] and we build on this work.

For generating primal solutions we use heuristics, similar in style to Goemans-Williamson [?], on the spectral bundle part PUP^T of the approximate primal solution $\tilde{X}$. One of the more successful variants pays special attention to the sign structure of the large eigenvalues. We improve these rounded solutions by simple local search techniques. The good quality of these solutions proved to be one major advantage of SDP over LP in our branch-and-cut comparisons.

After the addition of newly separated cutting planes there is no difficulty in restarting the bundle method from the old Lagrange multiplier solution by setting the new multipliers to zero. Extending the old subgradients to the new coordinates can be done easily, if the support of the new inequalities is restricted to the support of $\overline{W}$. This way the bundle model needs not be rebuilt in spite of the changes in dimension. Fortunately, no dramatic scaling problems seem to arise during the usual separation process, maybe because violation of the inequalities and changes in the multipliers seem to converge to zero at a common speed. Quite often, however, we observed significant scaling problems when the relaxation needs to be resolved after the addition of a branching constraint like setting $X_{ij} = 1$. Indeed, a few of the Lagrange multipliers – those associated with the new constraint $X_{ij} = 1$ and with the constraints containing the newly restricted edge – will typically change a lot, but most other multipliers seem not to move much. In this context the following idea for a scaling heuristic turned out to be quite effective. Take the two eigenvectors v, w to the two nonzero eigenvalues of X_{ij} and allow the more or the less change in the Lagrange multipliers in dependence on the Ritz values $v^T Av$ and $w^T Aw$ of each constraint matrix A.

In a combined bundle and cutting plane approach a heavy tailing off effect has to be expected and can be observed in solving the relaxation. To cope with this, we never wait for convergence but stop solving the relaxation quite early on several "lack of progress" criteria. Yet, the resulting rough approximations still need quite some time. Consequently the number of branch-and-bound nodes stays small and the common strong branching techniques of LP cannot be applied. To make up for this we developed a rather elaborate branching rule. Based on the vector labelling corresponding to PUP^T we try to cluster the nodes into subsets having well aligned vectors and investigate the effect on the cost function if two such clusters are forced to be aligned in the same or in the opposite direction. Given the two clusters where this shows the strongest effect, we pick a representative of each cluster, say nodes $\hat{\imath}$ and $\hat{\jmath}$, and set $X_{\hat{\imath}\hat{\jmath}}$ to $+1$ in one subproblem and to -1 in the other.

4 Computational Results

For our empirical investigations we used sparse graph instances from the samples presented in [?] varying in the number of edges between 1 500 and 500 000. We generally set $F = 0.525 f(V)$. As a branch-and-cut framework we use SCIP [?] with ILOG CPLEX 9.130 as LP-solver and the spectral bundle method developed by [?] as the SDP-solver. The computations were executed on a 3.2 GHz Pentium IV processor with 1 GB RAM.

In Table **??** we present a comparison of the performance of the cutting plane algorithms based on the knapsack tree inequalities with minimal roots (*kt*), capacity improved bisection knapsack walk inequalities (*bkw*) and cycle inequalities (*cy*) in combination with the linear and the semidefinite relaxation. Note that the cycle inequalities are part of the integer linear model (**??**), so they are separated in each setting of the LP relaxation. In the tests of Table **??** we only computed the root node of the branch-and-cut tree. We added inequalities of each class separately as long as violated cuts were found, the time limit of 4 hours was not exceeded and a heuristically computed upper bound was not yet proven to be optimal. Along with the lower bounds we report the best upper bounds known to us but not necessarily achieved in the same computations.

Within the linear relaxation, the knapsack tree inequalities have the biggest impact on the improvement of the lower bound. This may seem surprising in view of the fact that the knapsack walk inequalities subsume the knapsack tree inequalities; the reason is that, for speed, both separators start from a few seed nodes and then the minimal root strengthening of Theorem **??** boosts the performance of knapsack trees. Column *all* shows that it is worth to apply all separators in the linear case. The cycle separator alone achieves very poor lower bounds, so studying the bisection cut polytope P_{B} pays off when considering the linear relaxation of MB. The situation is stunningly different in the semidefinite case. Here, the pure semidefinite relaxation (*none*) yields already good lower bounds. For very large instances like *alut2292.494500* the separation routines only slow down the solution process and thus lead to worse bounds when the computing time is a major limiting factor. The best results are achieved when the separator for cycle inequalities is used exclusively. The bisection specific inequalities, i.e., the knapsack tree and bisection knapsack walk inequalities, yield roughly the same performance. These also improve the bound significantly but in comparison to cycle inequalities computation times are higher.

Based on the results of Table **??** the parameters of our branch-and-cut codes were set as follows. For the LP-relaxation the knapsack tree separator is given the highest priority and separation frequency, followed by the cycle and the bisection knapsack walk separators. For the semidefinite relaxation the cycle separator is the only separator. We limited computation time to 10 hours for each instance. The computations are presented in Tables **??** and **??**. By solving big instances (Table **??**) the SDP-relaxation outperforms the LP-relaxation with respect to both quality of dual bounds and computation time. In most cases we also observed that after a few seconds the value of the current SDP-bound exceeds the value of the current LP-bound and stays ahead throughout. However, the situation looks quite different when solving graph instances coming from compiler design as in [**?**]. For these instances the linear solver is far ahead of the semidefinite one with respect to computation time.

5 Conclusion

In this paper we considered the minimum graph bisection problem, a combinatorial problem for which linear and semidefinite relaxations are easy to derive. Using previous polyhedral studies presented in [**?**] we developed separation routines for valid inequalities to the bisection cut polytope P_{B} and incorporated them into a common branch-and-cut framework for linear and semidefinite relaxations. On the basis of large sparse instances coming from VLSI design we showed the good performance of the semidefinite approach versus the mainstream linear one.

Aknowledgments. This work was supported by the German Research Foundation (DFG) under grants HE 3524/2-1/2 and MA 1324/2-1/2.

Table 1. Lower bounds computed by the linear and the semidefinite relaxation in the root of the b&b tree. Results on VLSI design graphs [**?**].

graph $n.m$	linear relaxation				semidefinite relaxation					ub
	cy	cy+bkw	cy+kt	all	$none$	bkw	kt	cy	all	
diw681.1494	18.4	134.9	**136.3**	135.9	77.3	135.1	134.7	**140.6**	137.9	142
taq1021.2253	23.1	74.1	113.2	**113.9**	60.1	112.9	112.4	**116.8**	115.0	118
dmxa1755.3686	0.0	42.6	87.1	**91.1**	37.5	89.3	89.0	**92.9**	90.3	94
diw681.3104	34.8	238.4	744.4	**829.2**	630.7	954.6	935.0	**988.8**	969.4	1011
taq334.3763	75.5	111.8	324.8	**324.9**	234.0	**324.8**	320.5	317.9	324.7	342
diw681.6402	46.8	136.2	304.6	**306.3**	280.8	320.7	310.4	319.7	**322.7**	331
gap2669.6182	8.6	28.3	71.1	**73.7**	35.8	72.7	**74.0**	**74.0**	73.0	74
alut2292.6329	3.8	17.1	55.3	**59.8**	39.7	74.0	74.6	**76.2**	74.9	77
taq1021.5480	74.1	154.4	639.8	**689.9**	1122.0	1510.8	1469.9	**1540.6**	1538.9	1650
dmxa1755.10867	20.4	31.7	137.1	**138.6**	94.6	143.0	142.0	143.5	**143.7**	150
alue6112.16896	0.0	8.2	7.8	**31.0**	52.9	117.6	99.9	**135.1**	98.0	136
gap2669.24859	**55.0**	55.0	55.0	55.0	46.0	55.0	55.0	55.0	55.0	55
taq1021.31641	151.8	215.1	372.5	**374.6**	359.4	386.6	301.8	**398.6**	396.7	404
alut2292.494500	559.0	740.3	1571.3	**1966.2**	**53950**	53374	46405	51071	47007	67815
mean lower bd	23	77	162	**186**	184	282	270	**289**	279	
mean time	40	8389	11646	10267	120	5684	3386	1476	5684	

n number of nodes, m number of edges, ub upper bound

Table 2. Performance of the linear and the semidefinite branch-and-cut algorithms. Results on VLSI design graphs [**?**].

	linear relaxation					semidefinite relaxation				
graph $n.m$	*# b&b nodes*	*time*	*upper bound*	*lower bound*	*gap (%)*	*# b&b nodes*	*time*	*upper bound*	*lower bound*	*gap (%)*
diw681.1494	1686	10 h	144	140.8	2	237	10 h	142	140.5	1
taq1021.2253	95	10 h	118	118.0	0	1	322s	118	118.0	0
dmxa1755.3686	35	9 h	94	94	0	68	10 h	94	92.8	1
diw681.3104	1	10 h	1064	835.7	27	124	10 h	1011	1007.1	0.1
taq334.3763	351	4 h	342	342.0	0	2318	10 h	342	340.1	0.4
diw681.6402	3	10 h	357	315.2	13	159	10 h	331	329.2	0.1
gap2669.6182	1	4 h	74	74	0	1	651 s	74	74	0
alut2292.6329	1	10 h	77	69.5	10	96	10 h	77	76.1	1
taq1021.5480	1	10 h	2019	701.2	187	84	10 h	1650	1586.9	3
dmxa1755.10867	62	10 h	157	144.1	8	79	10 h	150	145.9	2
alue6112.16896	1	10 h	146	21.5	578	11	10 h	136	135.6	0
gap2669.24859	1	2525 s	55	55.0	0.0	1	491 s	55	55.0	0
taq1021.31641	1	10 h	426	375.3	13	9	10 h	404	399.0	1
alut2292.494500	1	*5 h	67815	1813.5	3639	1	5 h	67815	51880.0	30
geom. mean**	12	27144 s	208	156	8	42	17278 s	199	197	1

* Early termination due to a memory shortage, ** *alut2292.494500* excluded.

Table 3. Performance of the linear and the semidefinite branch-and-cut algorithms. Results on compiler design graphs [**?**].

	linear relaxation					semidefinite relaxation				
graph $n.m$	*# b&b nodes*	*time (sec.)*	*upper bound*	*lower bound*	*gap (%)*	*# b&b nodes*	*time (sec.)*	*upper bound*	*lower bound*	*gap (%)*
cb.30.47	354	1	266	266	0	25166	540	266	266	0
cb.30.56	326	2	379	379	0	10276	256	379	379	0
cb.45.98	49	5	989	989	0	2995	438	989	989	0
cb.47.101	100	3	527	527	0	4403	433	527	527	0
cb.47.99	12	5	765	765	0	963	113	765	765	0
cb.61.187	785	81	2826	2826	0	10333	*5907	2826	2647	7

* Early termination due to a memory shortage.

References

1. Achterberg, T.: Constraint integer programming. PhD-Thesis, PhD-Thesis,Technische Universität Berlin, Berlin (2007)
2. Armbruster, M., Fügenschuh, M., Helmberg, C., Martin A.: On the bisection cut polytope. Technical Report, Chemnitz/Darmstadt University of Technology (2007)
3. Barahona, F., Mahjoub, A. R. : On the cut polytope. Math.Prog. **36** (1986) 157–173
4. Conforti, M., Rao, M. R., Sassano, A.: The equipartition polytope I, II. Math. Prog. **49** (1990) 49–70, 71–90
5. de Souza, C. C.: The graph equipartition problem: Optimal solutions, extensions and applications. PhD-Thesis, Université Catholique de Louvain, Belgium (1993)
6. Deza, M., Laurent, M.: Geometry of Cuts and Metrics, volume **15** of Algorithms and Combinatorics, Springer (1997)
7. Eisenblätter, A.: Frequency Assignment in GSM Networks. PhD-Thesis,Technische Universität Berlin, Berlin (2001)
8. Ferreira, C. E., Martin, A., de Souza, C. C., Weismantel, R., Wolsey, L. A.: Formulations and valid inequalities for the node capacitated graph partitioning problem. Math. Prog. **74** (1996) 247–266
9. Garey, M. R., Johnson D. S.: Computers and Intractability. W.H. Freeman and Company, New York (1979)
10. Gilbert, J. R., Tarjan R. E.: The analysis of a nested dissection algorithm. Numer. Math. **50** (1979) 377–404
11. Goemans, M. X., Williamson, D. P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. ACM **42** (1995) 1115–1145
12. Helmberg, C.: A cutting plane algorithm for large scale semidefinite relaxations. In M. Grötschel, editor, *The Sharpest Cut*, MPS-SIAM Series on Optimization (2004) 233–256

13. Helmberg, C., Kiwiel, K. C.: A Spectral Bundle Method with Bounds. Math. Prog. **93** (2002) 173–194
14. Helmberg, C., Rendl, F.: A spectral bundle method for semidefinite programming. SIAM J. Optim. **10(3)** (2000) 673–696
15. Laurent, M., de Souza, C. C.: Some new classes of facets for the equicut polytope. Discr. App. Math. **62** (1995) 167–191
16. Johnson E., Mehrotra, A., Nemhauser, G.: Min-cut clustering. Math. Prog. **62** (1993) 133–152
17. Jünger, M., Martin, A., Reinelt, G., Weismantel, R.: Quadratic 0/1 optimization and a decomposition approach for the placement of electronic circuits, Math. Prog. B **63(3)**(1994) 257–279
18. Lengauer, T.: Combinatorial algorithms for integrated circuit layout. John Wiley and Sons Ltd, Chichester (1990)
19. Poljak, S., Rendl, F. : Nonpolyhedral relaxations of graph-bisection problems. SIAM J. Optim. **5(3)** (1995) 467–487
20. Rendl, F., Rinaldi, G., Wiegele, A.: A branch and bound algorithm for Max-Cut based on combining semidefinite and polyhedral relaxations. In Fischetti, M., Williamson, D.P. (eds.), IPCO 2007, LNCS **4513** (2007) 295–309
21. Weismantel, R.: On the 0/1 Knapsack polytope. Math. Prog. **77** (1997) 49–68