Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

ARIE M.C.A. KOSTER
ADRIAN ZYMOLKA

Polyhedral investigations on stable multi-sets

ZIB-Report 03–10 (July 2003)

Polyhedral investigations on stable multi-sets

Arie M.C.A. Koster[†]   Adrian Zymolka[†‡]

Abstract

Stable multi-sets are an evident generalization of the well-known stable sets. As integer programs, they constitute a general structure which allows for a wide applicability of the results. Moreover, the study of stable multi-sets provides new insights to well-known properties of stable sets. In this paper, we continue our investigations started in [9] and present results of three types: on the relation to other combinatorial problems, on the polyhedral structure of the stable multi-set polytope, and on the computational impact of the polyhedral results.

First of all, we embed stable multi-sets in a framework of generalized set packing problems and point out several relations. The second part discusses properties of the stable multi-set polytope. We show that the vertices of the linear relaxation are half integer and have a special structure. Moreover, we strengthen the conditions for cycle inequalities to be facet defining, show that the separation problem for these inequalities is polynomial time solvable, and discuss the impact of chords in cycles. The last result allows to interpret cliques as cycles with many chords.

The paper is completed with a computational study to the practical importance of the cycle inequalities. The computations show that the performance of state-of-the-art integer programming solvers can be improved significantly by including these inequalities.

In this paper, we continue our study of stable multi-sets started in [9]. We focus on polyhedral properties of the stable multi-set polytope as well as its linear relaxation, addressing both theoretical and computational aspects. Many of the results are motivated by known results for the stable set problem, thereby also providing more insights for the latter. As illustrative example, discussed in Section 3.3, the conditions under which chords do not influence the strength of cycle inequalities for stable multi-sets show why only odd holes yield facet defining inequalities for stable sets. Furthermore, the structure of stable sets can be identified as substructure in many integer programs, and hence the polyhedral properties of stable sets have been exploited by many software solutions for integer programming. As generalization, the stable multi-set structure may allow for similar exploitations of their polyhedral properties.

As introduction, we start this paper with a discussion of set packings and their generalizations in Section 1, thereby embedding stable multi-sets in a more general framework.

[†]Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, D-14195 Berlin, Germany.
[‡]Corresponding author. E-mail: zymolka@zib.de — URL: http://www.zib.de/zymolka/
Keywords: stable multi-sets, multi-set packing, polyhedral combinatorics
Mathematics Subject Classification (1991): 90C57, 90C10, 90C27

Correspondences and differences with stable sets guide this presentation. After definitions and preliminaries in Section 2, Section 3 is devoted to theoretical results on the stable multi-set polytope. We first characterize the extreme points of the linear relaxation and show their special structure. For the integer polytope, we strengthen the conditions under which the cycle inequalities are facet defining, and show that cycles with chords can be facet defining as well. Moreover, we provide a positive answer to the question stated in [9] whether a polynomial time algorithm for the separation of the cycle inequalities exists. Section 4 contains the results of a computational study on solving stable multi-set problems. We demonstrate the profitability of the cycle inequalities within a branch-and-cut framework. Finally, concluding remarks in Section 5 close the paper.

1 Stable multi-sets

Stable multi-sets have been introduced in [9] as an evident generalization of the well-known stable sets which are also referred to as independent sets, cocliques, or vertex packings. The last name already indicates that the stable set problem belongs to the fundamental class of set packing problems, which play an important role in graph theory and combinatorial optimization, cf. Schrijver [12]. To establish a theoretical foundation for stable multi-sets, we classify them as special case of a generalized set packing problem and discuss the relations between these problems.

1.1 Set packing and generalizations

Formally, *set packing* consists in the following task. Given a finite master set $M = \{1,\dots,m\}$ of items and a set $\mathcal{S} \subset 2^M$ of subsets, each with a weight $c_S \geq 0$, $S \in \mathcal{S}$, we seek a maximum weighted collection $\mathcal{T} \subseteq \mathcal{S}$ of pairwise disjunct subsets, i.e., $S_1 \cap S_2 = \emptyset$ $\forall\ S_1, S_2 \in \mathcal{T}, S_1 \neq S_2$. Each such collection $\mathcal{T}$ is called a *packing*. Clearly, each item is covered at most once by any packing, i.e., $|\{S \in \mathcal{T} \mid j \in S\}| \leq 1$ for all $j \in M$. For an extensive description of the problem and further relations, we refer to Borndörfer [1] and Schrijver [12].

By introducing binary variables x_S for all $S \in \mathcal{S}$, the set packing problem can be formulated as integer linear program by

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq \mathbb{1} \\ & x \in \{0,1\}^{\mathcal{S}} \end{aligned}$$

where c is the vector of subset weights, $\mathbb{1}$ denotes the vector of all 1's of appropriate size, and A is the item-subset incidence matrix, i.e., $a_{iS} = 1$ if $i \in S$, and 0 otherwise.

Two important specific set packing problems occur if A reflects relations between the vertices and edges of a graph G. For A being the vertex-edge incidence matrix (i.e., $|S| = 2$ for all $S \in \mathcal{S}$), the associated problem is known as *matching* problem, while the edge-vertex incidence matrix A (i.e., $|\{S \in \mathcal{S} \mid j \in S\}| = 2$ for all $j \in M$) yields the *stable set*

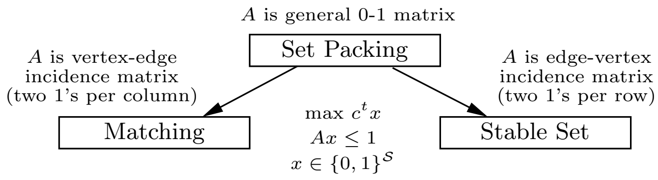

Figure 1: Scheme of relations between set packing problems.

problem. These problem relations are depicted in Figure 1. Note that in fact the set packing problem and the stable set problem are equivalent, since each set packing inequality can be interpreted as clique inequality for stable sets and split into the single edge inequalities, and vice versa.

A well-known generalization of matchings is given by *(integer) b-matchings* in which edges can be selected multiple times and vertices may be covered up to the bounds stated as b. So in the context of the integer program, the right hand side is replaced by b, and the variables can take general integer values. The corresponding integer program reads

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \in \mathbb{N}_0^{\mathcal{S}} \end{aligned}$$

where A is the vertex-edge incidence matrix, c contains the edge weights, and b indexed by the vertices represents the cover bounds. Stating additional bounds on the number of times the edges can be selected leads to *capacitated b-matchings*. An b-matching can also be interpreted as an edge multi-set, where a *multi-set* is a set with allowed repetition of elements and represented by its *multiplicity function* $x : E \to \mathbb{N}_0$.

The same kind of generalization can also be applied to general set packings: we allow to pick sets from $\mathcal{S}$ multiple times as long as the number of times items are packed does not exceed given bounds. In this case, a solution is a multi-set of the elements of $\mathcal{S}$. Therefore the resulting problem is denoted as *multi-set packing* (MSP). The construction also carries over to stable sets and yields the *stable multi-sets* (SMS). Both problems have the same integer program as b-matching, only varying in the incidence structure represented by A. Figure 2 illustrates these relations of the described problems. Remark that in the context of (strongly) t-perfectness of graphs, stable multi-sets are also known as b-stable sets [12]. For both SMS and MSP, we consider the capacitated version, i.e., for all variables an upper bound exists. For MSP, these bounds can be easily included in A. For SMS, stating that A has *at most* two 1's per row instead of *exactly* two 1's per row allows for a simple incorporation of vertex bounds (for stable sets, these are implicitly established by the $\mathbb{1}$'s vector).

From a graph-theoretic point of view, a similar interpretation as for b-matchings can be given for stable multi-sets: vertices can be selected multiple times, and the edges may be covered up to the stated bounds. The most general multi-set packing indeed requires the extension to a hypergraph where it represents multiple vertex selections not exceeding

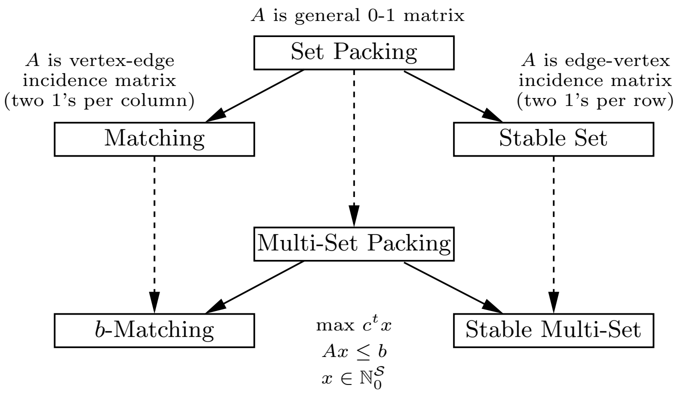

Figure 2: Extended scheme including problem generalizations.

bounds on the hyperedges (or multiple hyperedges depending on the interpretation of the matrix A).

1.2 Elementary relations

The relations illustrated in Figure 2 directly cause two questions:

(i) Are stable multi-sets and multi-set packings also equivalent?

(ii) Can stable multi-set problems easily be transformed to stable set problems?

The answer to the first question is negative as the following example makes clear.

Example 1.1 *Consider the MSP defined by three vertices $\{a, b, c\}$ and a single hyperedge covering all vertices with bound 2 as illustrated in Figure 3(a). Each multi-set of vertices up to a cardinality of 2 is a valid packing. This MSP cannot be written as an SMS on the same vertices. Since an arbitrary combination of the vertices up to 2 can be taken, the edge bounds all have to be at least 2, cf. Figure 3(b). However, this allows also for a solution like $x_a = x_b = x_c = 1$ which has cardinality 3.*

The equivalence of stable set and set packing is therefore ascribed to the case in which all bounds are 1.

The second question concerns the relation between stable sets and stable multi-sets. The following construction describes a transformation of stable multi-sets to stable sets that is pseudo-polynomial in the input size of the problem.

Consider a stable multi-set instance on a graph $G = (V, E)$ with vertex bounds α_v for all $v \in V$ and edge bounds β_{vw} for all $vw \in E$. A corresponding stable set instance $G' = (V', E')$ is constructed as follows. We replace each vertex $v \in V$ by a vertex set $\{v'_1, v'_2, \ldots, v'_{\alpha_v}\}$. We interpret choosing v'_i in the stable set as choosing i times v in the stable multi-set (with none v'_i chosen meaning that v does neither occur in the stable multi-set). Hence, at most one vertex v'_i can be chosen, and thus $\{v'_1, v'_2, \ldots, v'_{\alpha_v}\}$ induce a

$$x_a + x_b + x_c \leq 2$$

(a) MSP

$$x_a + x_b \leq 2$$
$$x_b + x_c \leq 2$$
$$x_a + x_c \leq 2$$

(b) SMS

Figure 3: Example for distinction of multi-set packing and stable multi-set.

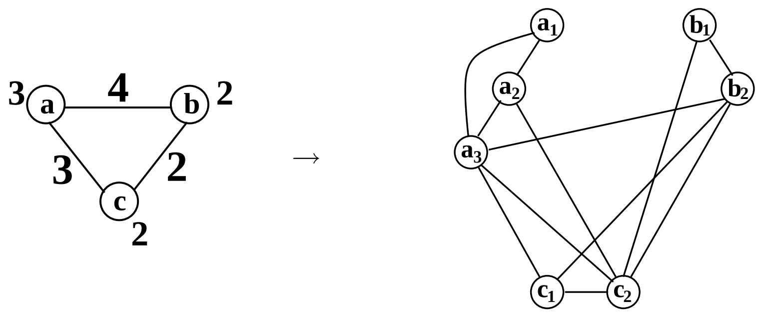

Figure 4: Transformation of a stable multi-set instance into an appropriate stable set instance.

clique in G'. Finally, for each edge $vw \in E$ with bound β_{vw} we insert all edges $v'_i w'_j$ with $i + j > \beta_{vw}$. The construction is illustrated in Figure 4 by an example. It is easy to verify that each stable set x' in G' corresponds to a stable multi-set x in G by $x_v = \sum_{i=1}^{\alpha_v} i \cdot x'_{v'_i}$. Unfortunately, the presented transformation significantly enlarges the graph to consider. For $\alpha = \max_{v \in V} \alpha_v$, $n = |V|$, and $m = |E|$, the resulting stable set instance has $\mathcal{O}(n\alpha)$ vertices and $\mathcal{O}((n + m)\alpha^2)$ edges. In addition, the solution correspondence does not allow for an easy adaption of known inequalities and other results for stable sets to the multi-set case by this transformation.

2 Definitions and preliminaries

In this paper, we use the following notation and refer to [12] for non-explained elementary graph theoretical notions. An undirected graph $G = (V, E)$ consists of a set of vertices V and a set of edges E. Throughout the paper, we use $n = |V|$ and $m = |E|$. We assume all considered graphs to be simple, i.e., contain no loops and no multiple edges. We always use the short notation vw for an edge $\{v, w\} \in E$. Let $N(v)$ denote the set of neighbors of $v \in V$, i.e., $N(v) := \{w \in V \mid vw \in E\}$. Moreover, for $W \subseteq V$, let $N(W) := \{v \in V \setminus W \mid vw \in E, w \in W\}$ be the set of vertices that W separates from the rest of the graph. Usually we denote a cycle by its vertex set C, and the associated edge set by E_C. Given a subset $S \subseteq V$ of vertices, the subgraph of G induced by S is denoted by $G[S]$. Similarly, S denotes the vector x restricted to the vertices in $S \subseteq V$. For $S \subseteq V$, let $e^S \in \{0,1\}^V$ denote the characteristic vector defined by $e^S_v = 1$ if $v \in S$ and $e^S_v = 0$ otherwise.

Throughout this paper, we use a graph-oriented definition of stable multi-sets. Let $G = (V,E)$ be a graph, $\alpha_v > 0$ and $c_v > 0$ integers associated with each vertex $v \in V$, and $\beta_{vw} > 0$ integers associated with each edge $vw \in E$. A stable multi-set (SMS) is a vertex multi-set defined by a multiplicity vector $x \in \mathbb{N}_0^V$ such that $0 \leq x_v \leq \alpha_v$ for all $v \in V$ and $x_v + x_w \leq \beta_{vw}$ for all $vw \in E$. The SMS problem is to find a stable multi-set of maximum value $\sum_{v \in V} c_v x_v$.

A number of reduction rules for the SMS problem has been stated in [9]. Without loss of generality, we assume in the sequel each SMS problem to be irreducible, i.e., $\max\{\alpha_v, \alpha_w\} \leq \beta_{vw} < \alpha_v + \alpha_w$ for all $vw \in E$, and $\min_{w \in N(v)}\{\beta_{vw} - \alpha_w\} = 0$ for all $v \in V$.

Given a triple G, α, β, the set of valid stable multi-sets is denoted by

$$T(G,\alpha,\beta) = \{x \in \mathbb{N}_0^V \,|\, 0 \leq x_v \leq \alpha_v \;\forall v \in V,\;\; x_v + x_w \leq \beta_{vw} \;\forall vw \in E\}.$$

The convex hull of this set is named as $T_{IP}(G,\alpha,\beta) = \text{conv}(T(G,\alpha,\beta))$. Moreover, $T_{LP}(G,\alpha,\beta)$ refers to the polytope described by the linear relaxation of $T(G,\alpha,\beta)$:

$$T_{LP}(G,\alpha,\beta) = \{x \in \mathbb{R}^V \mid 0 \leq x_v \leq \alpha_v \;\forall v \in V,\;\; x_v + x_w \leq \beta_{vw} \;\forall vw \in E\}.$$

If there is no danger of confusion, we use T, T_{IP}, and T_{LP} as short version of $T(G,\alpha,\beta)$, $T_{IP}(G,\alpha,\beta)$, and $T_{LP}(G,\alpha,\beta)$, respectively.

As a generalization of stable sets, the stable multi-set problem is obviously $\mathcal{NP}$-hard. In [9], we started the study of valid inequalities that describe facets of T_{IP}. First of all, the non-negativity inequality $x_v \geq 0$ defines a facet of T_{IP} for all $v \in V$, whereas the *vertex inequality* $x_v \leq \alpha_v$ is facet defining if and only if $\beta_{vw} > \alpha_v$ for all $w \in N(v)$. An *edge inequality* $x_v + x_w \leq \beta_{vw}$ defines a facet of T_{IP} if and only if for all $u \in N(\{v,w\})$, there exist integers $\bar{x}_v \leq \alpha_v$ and $\bar{x}_w \leq \alpha_w$ with $\bar{x}_v + \bar{x}_w = \beta_{vw}$, $\bar{x}_v < \beta_{vu}$ if $u \in N(v) \setminus \{w\}$, and $\bar{x}_w < \beta_{wu}$ if $u \in N(w) \setminus \{v\}$.

In addition to these classes of *model inequalities*, two other classes of inequalities have been identified. Given a cycle $C \subseteq V$ in G, the *cycle inequality* is defined as

$$\sum_{v \in C} x_v \leq \left\lfloor \tfrac{1}{2}\beta(C) \right\rfloor \tag{1}$$

where $\beta(C) = \sum_{vw \in E_C} \beta_{vw}$ is the sum of edge bounds on the cycle. The cycle inequality is redundant if $|C|$ is even or even-valued, i.e., $\beta(C)$ is even. Moreover, the inequality can also be redundant for odd-valued odd cycles. Non-redundancy holds if and only if

$$\min_{i=1,\ldots,2k+1} \left\{ \alpha_{v_i} + \sum_{p=1}^{k} \beta_{v_{i+2p-1}v_{i+2p}} \right\} > \left\lfloor \tfrac{1}{2}\beta(C) \right\rfloor \;. \tag{2}$$

In [9] also conditions which characterize facet defining cycle inequalities are provided; a strengthening of these conditions is presented in Section 3.2. Since the cuts of Chvátal-rank

1 are exactly the cycle inequalities, $T_{IP} = T_{LP}$ if and only if all cycles induce redundant inequalities. Recently, Gijswijt and Schrijver [6] proved that for graphs without bad K_4 subdivision[1], the system of vertex bounds, edge bounds, and cycle inequalities describes T_{IP} completely for arbitrary α and β.

The second class of inequalities proposed in [9] concerns so-called *β-cliques*. For a clique $Q \subseteq V$ in G with $\beta_{vw} = \beta$ for all $v, w \in Q$, $v \neq w$, the *β-clique inequality* is defined by

$$\sum_{v \in Q} x_v \leq |Q| \lfloor \tfrac{1}{2}\beta \rfloor + (\beta \bmod 2).$$

For $|Q| \geq 3$, the inequality defines a facet of T_{IP} if and only if $\alpha_v \geq \lceil \frac{1}{2}\beta \rceil$ for all $v \in Q$, β is odd, and for all $u \in N(Q)$, there exists $w \in Q$ with $w \notin N(u)$ or $\beta_{uw} \geq \lceil \frac{1}{2}\beta \rceil + 1$. Notice that not only maximal β-cliques can fulfill these conditions but also subcliques. In fact, if G is equivalent to a β-clique satisfying all conditions, T_{IP} is completely described by the model inequalities and the β-clique inequalities for all subcliques.

3 Polyhedral results

In this section, we discuss new polyhedral results for both T_{LP} and T_{IP}. In Section 3.1, we study properties of T_{LP}, whereas Sections 3.2–3.4 address different aspects of the cycle inequalities. Consecutively, we revisit the conditions under which cycle inequalities are redundant and facet defining, we study the influence of chords, and we present a polynomial time separation algorithm.

3.1 Extreme points of the linear relaxation

Nemhauser and Trotter [10] characterized the extreme points of the fractional stable set polytope. Based on this work, we investigate the polytope T_{LP} as well.

For the linear relaxation of the stable set polytope, it is known that all components of each extreme point only take values 0, $\frac{1}{2}$, or 1. Applying the same technique, we show for T_{LP} that all components of each extreme point always take half integer values:

Lemma 3.1 *Let x be an extreme point of $T_{LP}(G, \alpha, \beta)$. Then all component values are non-negative multiples of a half, i.e., for all $v \in V$ there exists $k_v \in \mathbb{N}_0$ with $x_v = \frac{1}{2}k_v$.*

Proof. Let x be an extreme point of $T_{LP}(G, \alpha, \beta)$ and define the index sets

$$\begin{aligned} U_{-1} &:= \{v \in V \mid \exists\, n \in \mathbb{N}_0 :\ n < x_v < n + \tfrac{1}{2}\}\ , \\ U_1 &:= \{v \in V \mid \exists\, n \in \mathbb{N}_0 :\ n + \tfrac{1}{2} < x_v < n + 1\}\ . \end{aligned}$$

[1] A K_4 subdivision is a graph that can be constructed by subdividing the edges of K_4. It is called *odd* if each triangle of the K_4 subdivision is odd. It is called *good*, if it is odd and if there are two disjoint edges of K_4 such that these are not subdivided and the other four edges are subdivided to even length paths. Finally a *bad* K_4 subdivision is a K_4 subdivision that is not good, see [5].

Assume $U_{-1} \neq \emptyset$ or $U_1 \neq \emptyset$. We then set

$$\varepsilon_1 := \tfrac{1}{2} \min \left\{ \min_{v \in U_{-1}} \{x_v - \lfloor x_v \rfloor\}, \ \min_{v \in U_1} \{\lceil x_v \rceil - x_v\} \right\} > 0$$

$$\varepsilon_2 := \tfrac{1}{2} \min \left\{ 1, \min_{\substack{vw \in E: \\ x_v + x_w < \beta_{vw}}} \{\beta_{vw} - x_v - x_w\} \right\} > 0$$

and $\varepsilon := \min\{\varepsilon_1, \varepsilon_2\} > 0$, and define two points y, z by

$$y_v := \begin{cases} x_v + k\varepsilon \ , & v \in U_k, \ k = \pm 1, \\ x_v \ , & \text{otherwise,} \end{cases} \qquad z_v := \begin{cases} x_v - k\varepsilon \ , & v \in U_k, \ k = \pm 1, \\ x_v \ , & \text{otherwise.} \end{cases}$$

By assumption, x, y, z are mutually different, and the construction yields $x, y, z \in T_{LP}(G, \alpha, \beta)$. But $x = \frac{y+z}{2}$ contradicts x to be an extreme point of $T_{LP}(G, \alpha, \beta)$. Hence, $U_{-1} = \emptyset = U_1$, which completes the proof. ■

The next natural step consists in a characterization of all extreme points of the linear relaxation. In case of stable sets, Nemhauser and Trotter [10] showed that x is an extreme point if and only if it can be written as $x = e^S + \frac{1}{2} e^T$, where S is a stable set, T induces a subgraph whose components are non-bipartite, and $(S \cup N(S)) \cap T = \emptyset$. Such a (de)composition of all extreme points is strongly related to the fact that all bounds are 1, which in particular implies $x_v = 0$ for all $v \in N(S)$. In the multi-set case, the edge bounds prohibit such an implication, and a composition has to take the edge bounds between S and T into account. However, the decomposition carries over, actually in two alternative ways stated in Corollary 3.5 and Proposition 3.6.

The first alternative distinguishes vertices with integer and fractional values. The following lemma is helpful in this context:

Lemma 3.2 *Let $x \in T_{LP}(G, \alpha, \beta)$ and $S := \{v \in V \mid x_v \neq \lfloor x_v \rfloor\}$. Then $x_v + x_w < \beta_{vw}$ for all $vw \in E$ with $v \in S$, $w \in N(S)$.*

Proof. Follows directly from the fact that x_v is fractional whereas x_w and β_{vw} are integer. ■

First, we focus on the vertices with fractional values and show that the point extremity carries over to a reduced stable multi-set instance induced by the associated part of G:

Lemma 3.3 *Let $x \in T_{LP}(G, \alpha, \beta)$ and $S := \{v \in V \mid x_v \neq \lfloor x_v \rfloor\}$. If x is an extreme point of $T_{LP}(G, \alpha, \beta)$, then $x[S]$ is an extreme point of $T_{LP}(G[S], \alpha, \beta)$. Moreover, each component of $G[S]$ contains an odd-valued odd cycle C that satisfies* (2)*.*

Proof. Suppose $x[S]$ is not an extreme point of $T_{LP}(G[S], \alpha, \beta)$. Then there exists $y[S] \neq 0$ such that $x[S] \pm y[S] \in T_{LP}(G[S], \alpha, \beta)$. By Lemma 3.2, $\beta_{vw} - x_v - x_w > 0$ for all $vw \in E$, $v \in S$, $w \in N(S)$. But then $x \pm \varepsilon y \in T_{LP}(G, \alpha, \beta)$ for some $\varepsilon \neq 0$ and y extended with zeros to the size of x. Hence, x is not an extreme point of $T_{LP}(G, \alpha, \beta)$, a contradiction. Thus, $x[S]$ is an extreme point of $T_{LP}(G[S], \alpha, \beta)$.

Now, let U induce a component of $G[S]$. Clearly, $x[U]$ is an extreme point of $T_{LP}(G[U], \alpha, \beta)$. Since $x[U]$ is fractional by construction, we know that $T_{LP}(G[U], \alpha, \beta) \neq T_{IP}(G[U], \alpha, \beta)$. By Corollary 2 from [9], this is equivalent to the case when $G[U]$ contains an odd-valued odd cycle that satisfies (2). ■

A similar property holds for the integer valued subgraph:

Lemma 3.4 *Let $x \in T_{LP}(G, \alpha, \beta)$ and $T := \{v \in V \mid x_v \text{ integer}\}$. If x is an extreme point of $T_{LP}(G, \alpha, \beta)$, then $x[T]$ is an integer extreme point of $T_{LP}(G[T], \alpha, \beta)$.*

Proof. By definition of T, $x[T]$ is integer. Suppose $x[T]$ is not an extreme point of $T_{LP}(G[T], \alpha, \beta)$. Then there exists $y[T] \neq 0$ such that $x[T] \pm y[T] \in T_{LP}(G[T], \alpha, \beta)$. Again, by Lemma 3.2, it holds that $x \pm \varepsilon y \in T_{LP}(G, \alpha, \beta)$ for some $\varepsilon \neq 0$ and y extended with zeros to the size of x. Hence, x is not an extreme point of $T_{LP}(G, \alpha, \beta)$. ■

Subsuming both properties yields the following decomposition of extreme points:

Corollary 3.5 *Each extreme point $x \in T_{LP}(G, \alpha, \beta)$ can be written as $x = x[S] + x[T]$, where $x[S]$ and $x[T]$ have the properties defined in Lemma 3.3 and 3.4.*

Unfortunately it is in general not possible to combine arbitrary extreme points $x[S]$ and $x[T]$ to an extreme point x of T_{LP}, since this ignores connecting edge bounds β_{vw} for $vw \in E$, $v \in S$, $w \in T$.

An alternative decomposition is presented in the following proposition:

Proposition 3.6 *Let $x \in \mathbb{R}^n_+$ and $S := \{v \in V \mid x_v \neq \lfloor x_v \rfloor\}$. If x is an extreme point of $T_{LP}(G, \alpha, \beta)$, then x can be written as $x = x' + \frac{1}{2}e^S$ with*

(i) x' an integer extreme point of $T_{LP}(G, \alpha', \beta')$ on the reduced instance with $\alpha'_v := \alpha_v - e^S_v$ for all $v \in V$ and $\beta'_{vw} := \beta_{vw} - e^S_v e^S_w$ for all $vw \in E$, and

(ii) each component of $G[S]$ contains an odd-valued odd cycle C that satisfies (2).

Proof. Let $x \in \mathbb{R}^n_+$ be an extreme point of $T_{LP}(G, \alpha, \beta)$ and define $x' = \lfloor x \rfloor$. Then $x = x' + \frac{1}{2}e^S$ by Lemma 3.1.

Since x is an extreme point, there exist n linearly independent inequalities (from the canonical description of the polytope) satisfied at equality by x. We show that all of them are also satisfied at equality by x'. For non-negativity inequalities, this is clear. Vertex (upper bound) inequalities which hold at equality need the associated vertex to take its upper bound value $\alpha_v = \alpha'_v$, so x' has the same value for this vertex. Finally, for the edge inequalities by the definition of β', the fractional part of x_v and x_w is subtracted from both the left and right hand side. As a result, there are n linearly independent inequalities satisfied at equality for x', yielding it to be an (integer) extreme point of the reduced instance $T_{LP}(G, \alpha', \beta')$.

The second part follows directly from Lemma 3.3. ■

Again, this decomposition is not invertible: vectors composed of an arbitrary integer extreme point x' of $T_{LP}(G, \alpha', \beta')$ and an arbitrary set S that has the appropriate properties does not necessarily lead to an extreme point of $T_{LP}(G, \alpha, \beta)$.

3.2 Facet defining cycle inequalities

In this section, we reconsider the cycle inequalities (1). Let $C \subseteq V$ be the vertex set of an odd cycle $C = \{v_1, v_2, \ldots, v_{2k+1}\}$ in G. In the sequel, we always interpret the vertex indices modulo $2k+1$ (in the range $1, \ldots, 2k+1$). From [9], we know that

(i) even-valued odd cycles are dominated by model inequalities and hence redundant;

(ii) an odd-valued odd cycle may also be dominated by model inequalities, which is the case if and only if (2) does not hold;

(iii) if restricting to the odd-valued odd cycle, i.e., $G = (C, E_C)$, the cycle inequality is facet defining for $T_{IP}(G, \alpha, \beta)$ if and only if (2) holds and

$$\max_{i=1,\ldots,2k+1} \left\{ \sum_{p=1}^{k} \beta_{v_{i+2p-1}v_{i+2p}} \right\} \leq \left\lfloor \tfrac{1}{2}\beta(C) \right\rfloor . \tag{3}$$

At first sight, the two latter properties do not exclude the existence of odd-valued odd cycles that define a non-redundant inequality which is not a facet of T_{IP}. To disprove this possibility, the main step consists in the following lemma which strengthens the characterization in (iii):

Lemma 3.7 *Let $G = (C, E_C)$ be an odd-valued odd cycle with $C = \{v_1, \ldots, v_{2k+1}\}$. If (3) is violated, then (2) is also violated.*

Proof. Let $m \in \{1, \ldots, 2k+1\}$ be an arbitrary index for which the maximum in (3) is taken, then the precondition reads (with $\beta(C)$ odd)

$$\sum_{p=1}^{k} \beta_{v_{m+2p-1}v_{m+2p}} > \left\lfloor \tfrac{1}{2}\beta(C) \right\rfloor = \tfrac{1}{2} \left(\sum_{p=1}^{2k+1} \beta_{v_p v_{p+1}} - 1 \right) .$$

By elementary transformations (including index subtitution), we get

$$\beta(C) > 2 \sum_{p=0}^{k} \beta_{v_{m+2p}v_{m+2p+1}} - 1$$

which can, since both sides are integer, be turned into

$$\beta(C) \geq 2 \sum_{p=0}^{k} \beta_{v_{m+2p}v_{m+2p+1}} = 2 \left(\beta_{v_m v_{m+1}} + \sum_{p=1}^{k} \beta_{v_{m+2p}v_{m+2p+1}} \right) .$$

Since $\beta(C)$ is odd and the right hand side is even, the left hand side can be rounded down after division by 2. Finally, $\alpha_{v_{m+1}} \leq \beta_{v_m v_{m+1}}$ implies

$$\begin{aligned} \left\lfloor \tfrac{1}{2}\beta(C) \right\rfloor \geq \beta_{v_m v_{m+1}} + \sum_{p=1}^{k} \beta_{v_{m+2p} v_{m+2p+1}} &\geq \alpha_{v_{m+1}} + \sum_{p=1}^{k} \beta_{v_{m+2p} v_{m+2p+1}} \\ &\geq \min_{i=1,\dots,2k+1} \left\{ \alpha_{v_i} + \sum_{p=1}^{k} \beta_{v_{i+2p-1} v_{i+2p}} \right\} \end{aligned}$$

as claimed. ■

This finally closes the gap and yields:

Corollary 3.8 *Let $G = (C, E_C)$ be an odd-valued odd cycle. Then the cycle inequality* (1) *is either redundant or facet defining.*

As a further consequence of Lemma 3.7, we can restate Proposition 4 in [9] as follows:

Proposition 3.9 *Let $C = \{v_1, \dots, v_{2k+1}\}$ be an odd cycle in G with $\beta(C)$ odd. Then the cycle inequality* (1) *defines a facet of $T_{IP}((C, E_C), \alpha, \beta)$ if and only if* (2) *is satisfied.*

Taking a look at the original proof of Proposition 4 in [9], the former condition (3) was necessary to guarantee that the $2k + 1$ uniquely determined integer points satisfying (1) at equality have non-negative entries. As Lemma 3.7 shows that the condition can be left out, it follows that these $2k + 1$ points are valid for $T_{IP}((C, E_C), \alpha, \beta)$ if and only if the inequality is non-redundant.

3.3 Chords in cycles

Proposition 3.9 specifies whether a cycle inequality is facet defining in case of $G = (C, E_C)$. Now, we turn to the case of arbitrary graphs. Obviously, the cycle inequality (1) is valid for each cycle in each graph, and the next natural step is to ask under which additional conditions it is facet defining in more general cases. A first step in this direction is to consider cycles with chords, i.e., graphs $G = (C, E)$ with $E_C \subset E$.

Again, we assume that the cycle vertices are consecutively indexed as $C = \{v_1, \dots, v_{2k+1}\}$, $k \in \mathbb{N}$. Any two different vertices $v_{j_1}, v_{j_2} \in C$ are connected by two paths on the cycle, one of them with an even number of edges, the other with an odd one. In what follows, we focus on the odd path connecting v_{j_1} and v_{j_2}, and assume without loss of generality $j_1 < j_2$ and $j_2 - j_1$ odd. Next, the edge bounds on the path are alternatingly added up in two sums, the first beginning with the first edge on the path and then taking each second one until the other vertex is reached, i.e.,

$$\beta^{odd+}_{j_1 j_2} := \sum_{p=0}^{\frac{j_2 - j_1 - 1}{2}} \beta_{v_{j_1+2p} v_{j_1+2p+1}} \ ,$$

and the second by taking all other (intermediate) path edges, i.e.,

$$\beta^{odd-}_{j_1 j_2} := \sum_{p=1}^{\frac{j_2-j_1-1}{2}} \beta_{v_{j_1+2p-1} v_{j_1+2p}} .$$

For graphs consisting of an odd cycle with chords, it turns out that the cycle inequality remains facet defining if the chord bounds satisfy a simple condition on these sums:

Theorem 3.10 *Let $G = (C, E)$ be an odd-valued odd cycle $C = \{v_1, \dots, v_{2k+1}\}$ together with a single chord $e = v_{j_1} v_{j_2}$, $3 \le j_2 - j_1$ odd. Then the cycle inequality* (1) *defines a facet of $T_{IP}(G, \alpha, \beta)$ if and only if condition* (2) *is satisfied, and the chord bound satisfies*

$$\beta_{v_{j_1} v_{j_2}} \ge \beta^{odd+}_{j_1 j_2} - \beta^{odd-}_{j_1 j_2} + 1 . \tag{4}$$

Proof. From the proof of Proposition 4 in [9], we know that there are exactly $2k + 1$ uniquely determined points $x^1, \dots, x^{2k+1}$ satisfying the cycle inequality at equality. These points are given by

$$x^i_{v_j} = \tfrac{1}{2}(\beta^1_j - \beta^2_j + (-1)^{j-i}) ,$$

where $j - i$ is taken modulo $2k + 1$ (in the range $1, \dots, 2k + 1$) and

$$\beta^1_j = \sum_{p=0}^{k} \beta_{v_{j+2p} v_{j+2p+1}} \quad \text{and} \quad \beta^2_j = \sum_{p=1}^{k} \beta_{v_{j+2p-1} v_{j+2p}} .$$

So, the cycle inequality (1) defines a facet if and only if all these points are feasible for $T_{IP}(G, \alpha, \beta)$, i.e., satisfy all vertex and edge bounds on the cycle as well as satisfy the chord bound. By Proposition 3.9, the condition on the cycle is equivalent to (2). Thus it remains to show that the condition on the chord bound is equivalent to (4).

Clearly, all points $x^1, \dots, x^{2k+1}$ satisfy the chord bound if and only if

$$\begin{aligned} \beta_{v_{j_1} v_{j_2}} &\ge \max_{i=1,\dots,2k+1} \left\{ x^i_{v_{j_1}} + x^i_{v_{j_2}} \right\} \\ &= \max_{i=1,\dots,2k+1} \left\{ \tfrac{1}{2}(\beta^1_{j_1} - \beta^2 j_1 + (-1)^{j_1-i}) + \tfrac{1}{2}(\beta^1 j_2 - \beta^2 j_2 + (-1)^{j_2-i}) \right\} \\ &= \tfrac{1}{2}(\beta^1 j_1 - \beta^2 j_1 + \beta^1 j_2 - \beta^2 j_2) + \tfrac{1}{2} \max_{i=1,\dots,2k+1} \left\{ (-1)^{j_1-i} + (-1)^{j_2-i} \right\} \\ &= \beta^{odd+} j_1 j_2 - \beta^{odd-} j_1 j_2 + \tfrac{1}{2} \max_{i=1,\dots,2k+1} \left\{ (-1)^{j_1-i} + (-1)^{j_2-i} \right\} . \end{aligned}$$

To evaluate the last maximum, note that for $i = j_1 + 1$, $j_2 - i$ is even and $j_1 - i = 2k$ (computed modulo $2k + 1$), and hence also even. Thus, for this index i both exponents are even (modulo $2k + 1$), which yields $\max_{i=1,\dots,2k+1} \left\{ (-1)^{j_1-i} + (-1)^{j_2-i} \right\} = 2$, completing the proof. ■

Note that for the special case of stable sets, condition (4) is always violated: the chord bound is 1, whereas the right hand side always evaluates to 2. Thus, only odd holes may yield facet defining cycle inequalities for stable sets, which was proven by Padberg [11].

Of course, Theorem 3.10 is also the key to cycles with several chords as pointed out in the next corollary:

Corollary 3.11 *Let $G = (C, E)$ be a graph with $|C|$ odd and the edges $E_C \subset E$ define an Hamiltonian circuit in G with minimum total edge bound sum β_C. Then the inequality*

$$\sum_{v \in C} x_v \leq \left\lfloor \tfrac{1}{2} \beta_C \right\rfloor$$

is valid for T_{IP} and defines a facet if and only if β_C is odd, condition (2) *holds, and condition* (4) *is satisfied for all chords $vw \in E \setminus E_C$.*

So, in general also inequalities for odd cycles with chords can define facets of the stable multi-set polytope and hence are of interest for efficient algorithms. In contrast to stable sets, it does not suffice for strongly t-perfect graphs to add only odd hole inequalities to the model inequalities to get a complete description of $T_{IP}(G, \alpha, \beta)$. Moreover, Corollary 3.11 turns out to be valuable in an unexpected way. In [9], we stated the open problem to find a right hand side for general clique inequalities, i.e., cliques with non-uniform β values. The idea now is: *Cliques are just cycles with many chords.* So, the minimum Hamiltonian circuit always provides a valid right hand side for the clique inequality, which indeed defines a facet if the conditions in Corollary 3.11 apply. If they do not, but only one of the points $x^1, \ldots, x^{2k+1}$ is valid, β_C is the best right hand side for this clique. This partly answers the open question.

3.4 Separation of cycle inequalities

To strengthen the linear relaxation of the stable multi-set polytope, the inclusion of cycle inequalities is beneficial from both a theoretical and practical point of view. However, the number of cycle inequalities to be taken into account can be exponentially large, and thus it is not recommended to add all those inequalities to the linear program. Instead, we aim at separating violated cycle inequalities over the stable multi-set polytope. This separation problem reads:

Stable multi-set cycle separation
Instance: A stable multi-set problem instance (G, α, β) and $x \in T_{LP}$.
Question: Does there exist an odd-valued odd cycle C in G violating (1), i.e., with

$$\sum_{v \in C} x_v > \left\lfloor \tfrac{1}{2} \beta(C) \right\rfloor ?$$

As pointed out in [9], there is a polynomial time separation algorithm for cycle inequalities for the stable set problem proposed by [7]. In the following, we provide a generalization of this algorithm for the polynomial time separation of cycle inequalities in the stable multi-set case.

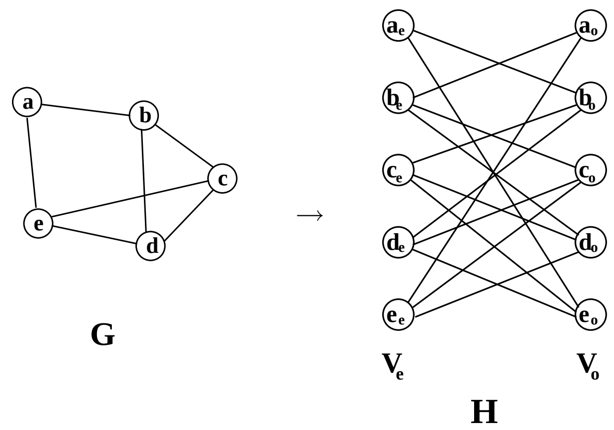

Figure 5: Construction of the auxiliary graph H for the separation of cycle inequalities for stable sets.

First, we recall the stable set case. For a fractional solution $x \in T_{LP}$, an auxiliary graph $H = (W, F)$ is constructed by $W = \{v_e, v_o \mid v \in V\}$ and $F = \{v_e w_o,\, w_e v_o \mid vw \in E\}$, i.e., H consists of two copies V_e, V_o of V and two copies of each edge $e \in E$ connecting the associated vertices from both vertex sets (see Figure 5). Moreover, each edge $vw \in E$ is assigned the value $z_{vw} = 1 - x_v - x_w$, and the edges in H overtake this value from the corresponding edge in G, i.e., $z_{v_e w_o} = z_{w_e v_o} = z_{vw}$ for all $vw \in E$.

Clearly, H is bipartite, thus any path from V_e to V_o has an odd number of edges. Since v_e and v_o correspond to the same vertex $v \in V$, any path p from v_e to v_o in H indicates an odd cycle C_p through v in G. Note that C_p need not to be simple even if p is, since p can visit both vertices u_e, u_o in H for a vertex $u \in V \setminus \{v\}$, and consequently C_p contains u twice. But in any case, C_p is odd and thus can be decomposed into even cycles and at least one simple odd cycle C (not necessarily containing v).

For separating cycle inequalities, a shortest path from v_e to v_o in H is computed for any $v \in V$. By the construction of H and the weights z, such a path with total weight smaller than 1 indicates a simple odd cycle C in G (possibly, by the decomposition mentioned above, as subcycle of the cycle C_p corresponding to p) for which the cycle inequality is violated, since

$$\sum_{v \in C} x_v = \tfrac{1}{2} \sum_{vw \in E_C} (x_v + x_w) = \tfrac{1}{2} \sum_{vw \in E_C} (1 - z_{vw}) = \tfrac{1}{2}(|C| - \sum_{vw \in E_C} z_{vw}) > \tfrac{1}{2}(|C| - 1) = \left\lfloor \tfrac{1}{2}|C| \right\rfloor \,.$$

The main idea behind this construction is that for a path starting in V_e, the index of any reached vertex indicates whether the path so far has an even or odd number of edges, or in other words, any path from v_e to v_o in H has odd length.

For stable multi-sets, this construction can be extended to reflect that both values have to be odd, the path length and the sum of its edge bounds. Instead of two copies of the vertices, four copies are introduced for odd/even-valued odd/even paths. These copies can be indexed V_{ee}, V_{eo}, V_{oe}, and V_{oo}, where for a path starting in V_{ee}, the first index indicates whether the number of edges of the path so far is even or odd, and the second index does

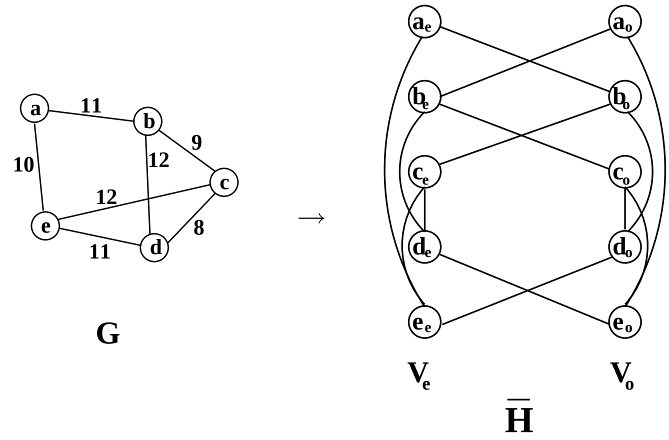

Figure 6: Construction of the auxiliary graph $\bar{H}$ for the separation of cycle inequalities for stable multi-sets.

the same for the path edge bound sum. In addition, each edge $vw \in E$ is copied four times connecting the appropriate vertices depending on the parity of β_{vw}. Finally, these edge copies overtake the weight $z_{vw} = \beta_{vw} - x_v - x_w$ from their original $vw \in E$. As result, a path from v_{ee} to v_{oo} in this bipartite graph represents a (not necessarily simple) odd-valued odd cycle. For paths with total weight smaller than 1, it can be shown that this cycle contains at least one simple odd-valued odd cycle with total weight smaller than 1 which again corresponds to a violated cycle inequality. An approach very similar to this construction has been independently developed by Cheng and de Vries [2].

The separation of cycle inequalities for stable multi-sets however can also be viewed from another perspective resulting in a shortest path computation on a graph with only $2n$ vertices. Recall that from the stable multi-set perspective, each edge in a stable set instance has edge bound 1. Hence, a path from v_e to v_o in H is not only a path of odd length, but also of odd edge bound sum. By introducing edges among the vertices in V_e and V_o for edges with even bound, this construction can be generalized to the stable multi-set case. Formally, we consider the auxiliary graph $\bar{H} = (\bar{W}, \bar{F})$ defined by

$$\begin{aligned} \bar{W} &= \{v_e, v_o \mid v \in V\}, \\ \bar{F} &= \{v_e w_e, v_o w_o \mid vw \in E,\ \beta_{vw} \text{ even}\} \cup \{v_e w_o, v_o w_e \mid vw \in E,\ \beta_{vw} \text{ odd}\}\ . \end{aligned}$$

This construction is exemplary depicted in Figure 6. For each edge $vw \in E$ in G, we define a weight $\bar{z}_{vw} = \beta_{vw} - x_v - x_w \geq 0$, which is carried over to the associated edges in $\bar{H}$.

Obviously, each violated odd-valued odd cycle inequality translates to a path from v_e to v_o with total $\bar{z}$-weight smaller than 1. The following lemma shows that odd-valued even cycles do not:

Lemma 3.12 *Let $x \in T_{LP}$ and C be an odd-valued even cycle in G. Then* $\sum_{vw \in E_C} \bar{z}_{vw} \geq 1$.

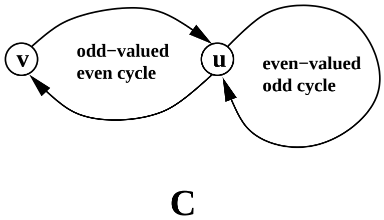

Figure 7: A non-simple odd-valued odd cycle C_p can decompose in an odd-valued even cycle and an even-valued odd cycle.

Proof. Let C be an odd-valued even cycle. Then

$$\sum_{vw \in E_C} \bar{z}_{vw} = \sum_{vw \in E_C} (\beta_{vw} - x_v - x_w) = \beta(C) - 2 \sum_{v \in C} x_v \geq \beta(C) - 2 \left\lfloor \tfrac{1}{2}\beta(C) \right\rfloor = 1.$$

Here the $\geq$ sign holds since the cycle inequalities are redundant for T_{LP} for even cycles. ■

Hence, each such path p corresponds to an associated odd-valued odd cycle C_p in G, but this cycle need not to be simple as mentioned before. Similar as for stable sets, non-simple cycles can be decomposed into simple cycles, but this time we have to be more careful, due to the doubled parity:

Proposition 3.13 *Let p be a path from v_e to v_o in $\bar{H}$ with total $\bar{z}$-weight smaller than 1. Then C_p contains at least one simple odd-valued odd cycle (not necessarily containing v) for which the cycle inequality* (1) *is violated.*

Proof. Each path p from v_e to v_o in $\bar{H}$ is odd-valued by construction. By Lemma 3.12, this path is also odd. If C_p is simple, we have found an odd-valued odd cycle for which the inequality (1) is violated. It remains to show that in case of a non-simple cycle, C_p contains at least one simple odd-valued odd cycle with total $\bar{z}$-weight smaller than 1.

Assume that C_p does not contain a simple odd-valued odd cycle. Then C_p decomposes into at least one simple odd-valued even cycle and one simple even-valued odd cycle, cf. Figure 7. However, from Lemma 3.12 we know that the total $\bar{z}$-weight of each odd-valued even cycle adds up to at least 1, a contradiction. Hence, C_p contains at least one simple odd-valued odd cycle. Since $\bar{z}_{vw} \geq 0$ for all $vw \in E$, the total $\bar{z}$-weight of this cycle remains smaller than 1, and thus implies a violated inequality (1). ■

Note that Proposition 3.13 in addition states that the class of odd-valued odd circuit (i.e., non-simple cycle) inequalities separated by Cheng and de Vries [2] in fact reduces to the class of odd-valued odd cycle inequalities. For the latter, we directly get:

Theorem 3.14 *For the stable multi-set problem, cycle inequalities can be separated in polynomial time.*

Proof. Let C be an odd-valued odd cycle for which the inequality (1) is violated. For $v \in C$, the shortest path from v_e to v_o in $\bar{H}$ has total weight smaller than 1. This path implies a violated cycle inequality (1) (not necessarily the one implied by C).

Hence, n shortest path computations detect violated cycle inequalities as long as some exist. This procedure clearly takes polynomial time. ■

Note that the polynomial time separation algorithm presented in the proof above heavily depends on the fact that $x \in T_{LP}$. If $x \notin T_{LP}$, Lemma 3.12 cannot be applied anymore, and C_p could indeed decompose into an odd-valued even cycle and an even-valued odd cycle.

Finally, [9] points out that the stable multi-set polytope is completely described by the model and all cycle inequalities for strongly t-perfect graphs which has been shown by Gerards and Schrijver [4]. Recently, Gijswijt and Schrijver [6] showed that for the superclass of graphs that does not contain a bad K_4 subdivision, T_{IP} has Chvátal rank at most 1. Together with Theorem 3.14, we conclude:

Corollary 3.15 *The stable multi-set problem is polynomial time solvable for graphs without a bad K_4 subdivision.*

4 Computational results

In this section, we report on computational studies on the impact of the valid inequalities known for stable multi-sets. Since the β-clique inequalities turned out to be of minor computational importance, we focus on the class of cycle inequalities. We first describe the setting and the instances, before we present the results of two comparisons on the benefit of separating odd-valued odd cycles.

4.1 Setting and instances

To evaluate the impact of cycle inequalities, we implemented a branch-and-cut algorithm for the stable multi-set problem with C++ as programming language. ILOG's Concert Technology has been used as a general framework for the implementation of the branch-and-cut algorithm, together with CPLEX, version 7.5 [8] as (integer) linear programming solver. All computations have been carried out on a PC with a 2.53 GHz Intel Pentium 4 processor, 2 GB Internal Memory, and Linux as operating system.

For this computational study, we adapted stable set instances to stable multi-sets. Since a maximum stable set corresponds to a maximum clique in the complement of the graph, the so-called DIMACS maximum clique instances [3] are frequently used for computational studies on stable sets. This set contains 66 graphs ranging from 28 upto 3361 vertices (cf. Table 1 for the exact sizes of the graphs). Stable multi-set instances have been generated from these instances in four steps:

(i) complement the graph,

(ii) randomly generate values $\alpha_v \in \{5, 6, \ldots, 15\}$ for all vertices $v \in V$,

(iii) randomly generate values $\beta_{vw} \in \{\max\{\alpha_v, \alpha_w\}, \ldots, \alpha_v + \alpha_w - 1\}$ for all $vw \in E$,

(iv) apply reduction rules to generate irreducible instances.

Note that by steps (ii) and (iii) most reduction rules of [9] have already been considered. Only the reduction of α_v by $\min_{w \in N(v)}(\beta_{vw} - \alpha_w)$ could lead to lower bounds and graph modifications. However, for the used instances no vertices have been removed by this reduction. Moreover, we set $c_v = 1$ for all $v \in V$ in our study, i.e., we solve the maximum cardinality stable multi-set problem.

4.2 Computational comparison

We report on two comparisons which show the potential of cycle inequalities. First of all, we compute the value of the *linear programming relaxation* with and without cycle inequalities, indicating the progress towards the solution value. Our second comparison concerns the performance of some integer programming algorithms to find *integer solutions*.

LP relaxation By the addition of cycle inequalities to the linear relaxation, the gap between LP and IP can be reduced substantially. In fact, from Corollary 3.15 we know that this gap can be closed completely for graphs without a bad K_4 subdivision. To test their impact in general, we have computed the LP value before and after the separation of the cycle inequalities. The results are presented in Table 1. Here, z_{LP} refers to the value of the linear relaxation, z_{LP}^+ to the LP value including the cycle inequalities, z_{IP} to the value of the optimal integer solution (or best known solution in case the optimal solution is not known), and the column "gap closed" refers to the percentage by which the gap between LP value and IP value is closed due to the inserted cycle inequalities, i.e., reflects the value $\frac{z_{LP}-z_{LP}^+}{z_{LP}-z_{IP}}$. Finally, the columns "# ineq." and "# rnds" list respectively the total number of inequalities separated and the number of separation rounds, i.e., the number of times the LP has been resolved until no violated inequalities could be found anymore.

For six instances, no violated cycle inequalities could be found, and thus we can conclude that for these instances the LP solution is integral. For five other instances, some violated inequalities have been generated although the optimal solution value is already attained by the LP solution. Moreover, Table 1 shows that for 37 of the 55 remaining instances the gap is completely closed by the cycle inequalities. In these cases, the number of separation rounds is typically small. For the remaining instances, the gap is closed by 85% on average with a minimum of 57%. Altogether, we can conclude that the cycle inequalities are indeed effective to improve the LP relaxation of the stable multi-set polytope.

Integer solutions Our second comparison is on the performance of the MIP solver with and without odd-valued odd cycle separation. Herefore, a total of three scenarios has been considered:

BB represents the usual branch-and-bound method in which no cycle inequalities are separated at all;

CPBB denotes the method in which cycle inequalities are separated only in the root node of the search tree, then continuing with branch-and-bound to explore the tree;

BC applies the branch-and-cut method, separating cycle inequalities in each node of the search tree.

instance	n	m	z_{LP}	z^+_{LP}	z_{IP}	gap closed	# ineq.	# rnds
MANN-a9	45	72	185.5	184.00	184	100.00	3	2
MANN-a27	378	702	1516.0	1500.00	1500	100.00	32	2
MANN-a45	1035	1980	4240.5	4199.00	4199	100.00	92	2
MANN-a81	3321	6480	13297.0	13164.00	13164	100.00	274	2
brock200-1	200	5066	923.5	923.00	923	100.00	43	2
brock200-2	200	10024	957.0	955.00	955	100.00	136	3
brock200-3	200	7852	969.5	969.00	969	100.00	3	2
brock200-4	200	6811	948.0	948.00	948	—	65	2
brock400-1	400	20077	1928.0	1927.00	1927	100.00	11	2
brock400-2	400	20014	1935.0	1934.00	1934	100.00	42	3
brock400-3	400	20119	1937.0	1930.00	1930	100.00	636	11
brock400-4	400	20035	1945.0	1945.00	1945	—	1	2
brock800-1	800	112095	3871.0	3856.00	3856	100.00	817	14
brock800-2	800	111434	3811.5	3798.25	3798	98.15	608	11
brock800-3	800	112267	3813.0	3795.76	3795	95.78	730	10
brock800-4	800	111957	3874.5	3861.31	3861	97.70	451	11
c-fat200-1	200	18366	974.5	973.00	973	100.00	33	2
c-fat200-2	200	16665	968.5	967.00	967	100.00	54	5
c-fat200-5	200	11427	930.0	930.00	930	—	44	2
c-fat500-1	500	120291	2415.0	2394.06	2391	87.25	445	8
c-fat500-2	500	115611	2350.0	2329.08	2324	80.46	601	13
c-fat500-5	500	101559	2369.5	2347.33	2344	86.94	845	13
c-fat500-10	500	78123	2412.0	2398.33	2398	97.64	566	12
hamming6-2	64	192	257.0	257.00	257	—	0	1
hamming6-4	64	1312	301.0	301.00	301	—	1	2
hamming8-2	256	1024	1187.0	1187.00	1187	—	0	1
hamming8-4	256	11776	1255.5	1252.00	1252	100.00	232	8
hamming10-2	1024	5120	4593.0	4593.00	4593	—	0	1
hamming10-4	1024	89600	4864.0	4850.00	4850	100.00	1086	8
johnson8-2-4	28	168	127.0	127.00	127	—	0	1
johnson8-4-4	70	560	336.0	335.00	335	100.00	2	2
johnson16-2-4	120	1680	600.0	599.00	599	100.00	45	3
johnson32-2-4	496	14880	2468.0	2467.00	2467	100.00	139	2

continued on next page

Table 1: Improvement of the LP value by cycle inequalities.

In order to have a fair comparison, the scenarios have been run with the default CPLEX parameters. Only a computation time limit of two hours (for the solution including separation) has been set. Note that the CPLEX default MIP parameters include the automatic separation of Gomory and other cuts.

As already pointed out by Table 1, for many instances the LP (with/without cycle inequalities) is already integral. Therefore, we subdivided the set of instances into three subsets according to their difficulty:

(i) instances for which the integer optimal solution is already found in the root node by BB (28 graphs, including 11 for which $z_{LP} = z_{IP}$),

instance	n	m	z_{LP}	z_{LP}^{+}	z_{IP}	gap closed	# ineq.	# rnds
continued from previous page								
keller4	171	5100	846.0	845.00	845	100.00	75	3
keller5	776	74710	3693.5	3689.00	3689	100.00	306	3
keller6	3361	1026582	15966.5	15815.41	15708*	58.45	7642	17
p-hat300-1	300	33917	1469.0	1465.00	1465	100.00	135	3
p-hat300-2	300	22922	1486.5	1481.00	1481	100.00	211	4
p-hat300-3	300	11460	1429.5	1429.00	1429	100.00	18	2
p-hat500-1	500	93181	2320.0	2298.30	2297	94.35	619	10
p-hat500-2	500	61804	2329.5	2321.00	2321	100.00	381	4
p-hat500-3	500	30950	2406.0	2403.00	2403	100.00	339	7
p-hat700-1	700	183651	3350.5	3322.21	3317	84.45	645	10
p-hat700-2	700	122922	3348.5	3334.47	3334	96.76	632	12
p-hat700-3	700	61640	3364.0	3357.00	3357	100.00	998	3
p-hat1000-1	1000	377247	4720.5	4661.26	4642*	75.46	1581	11
p-hat1000-2	1000	254701	4766.0	4730.46	4728	93.53	1104	12
p-hat1000-3	1000	127754	4816.5	4798.06	4797	94.56	1297	14
p-hat1500-1	1500	839327	7040.0	6924.81	6838*	57.02	3323	14
p-hat1500-2	1500	555290	7031.5	6949.24	6923*	75.82	3083	13
p-hat1500-3	1500	277006	7183.5	7148.97	7144	87.42	1771	13
san200-0.7-1	200	5970	978.5	978.00	978	100.00	12	2
san200-0.7-2	200	5970	948.0	947.00	947	100.00	97	2
san200-0.9-1	200	1990	952.0	952.00	952	—	0	1
san200-0.9-2	200	1990	927.0	926.33	926	67.00	8	2
san200-0.9-3	200	1990	934.0	933.00	933	100.00	97	3
san400-0.5-1	400	39900	1902.5	1899.00	1899	100.00	267	4
san400-0.7-1	400	23940	1946.0	1945.00	1945	100.00	220	3
san400-0.7-2	400	23940	1897.0	1896.00	1896	100.00	24	2
san400-0.7-3	400	23940	1959.0	1958.00	1958	100.00	35	2
san400-0.9-1	400	7980	1878.0	1878.00	1878	—	0	1
san1000	1000	249000	4740.5	4703.00	4703	100.00	1268	8
sanr200-0.7	200	6032	952.0	951.00	951	100.00	97	4
sanr200-0.9	200	2037	928.0	928.00	928	—	107	2
sanr400-0.5	400	39816	1968.5	1968.00	1968	100.00	74	2
sanr400-0.7	400	23931	1965.5	1965.00	1965	100.00	22	2

* Value of best known solution.

Table 1: Improvement of the LP value by cycle inequalities.

(ii) intances with an integer optimal solution found in the root node by CPBB (28 graphs), and

(iii) all remaining instances (10 graphs).

For the first set of instances, the automatic separation of Gomory cuts and the CPLEX-internal heuristic already solve the problem. Therefore, these instances are left out in our further considerations.

For the second set of instances, the results of the comparison between BB and CPBB are

instance	BB					CPBB			
	time (sec.)	nodes	left	value	gap	time (sec.)	sep. time (sec.)	# ineq.	value
MANN-a81	7371.41	1969753	1932849	13164	0.40	2.44	2.15	274	13164
brock200-2	5.79	18	0	955	0	1.39	0.47	132	955
brock400-2	3.33	1	0	1934	0	4.25	1.96	44	1934
brock400-3	55.14	24	0	1930	0	33.53	9.44	828	1930
brock800-1	752.27	1926	0	3856	0	211.63	97.16	766	3856
brock800-2	356.50	1035	0	3798	0	210.46	86.48	541	3798
brock800-3	1674.50	7784	0	3795	0	310.18	94.53	710	3795
brock800-4	288.69	447	0	3861	0	189.58	135.14	460	3861
c-fat200-1	3.36	5	0	973	0	1.40	0.35	33	973
c-fat200-2	3.24	5	0	967	0	1.62	0.48	44	967
c-fat500-10	445.22	1617	0	2398	0	72.39	33.60	505	2398
hamming8-4	4.80	5	0	1252	0	4.55	2.08	238	1252
hamming10-4	7285.03	52415	1332	4850	0.02	282.74	90.64	1007	4850
johnson32-2-4	6.39	1	0	2467	0	2.69	0.95	120	2467
keller4	0.88	2	0	845	0	0.53	0.26	70	845
keller5	57.63	8	0	3689	0	38.91	9.73	296	3689
p-hat300-1	6.39	2	0	1465	0	5.76	1.82	110	1465
p-hat300-2	29.44	36	0	1481	0	7.27	2.04	260	1481
p-hat500-1	7282.35	40059	40006	2294	0.22	167.96	71.62	730	2297
p-hat500-2	135.13	44	0	2321	0	40.77	18.06	350	2321
p-hat500-3	85.72	12	0	2403	0	28.04	17.01	342	2403
p-hat700-2	467.38	1573	0	3334	0	245.51	131.10	544	3334
p-hat700-3	259.44	8	0	3357	0	120.79	20.27	1020	3357
p-hat1000-3	2436.63	7670	0	4797	0	1430.24	192.46	1560	4797
san200-0.9-3	0.99	5	0	933	0	0.28	0.15	100	933
san400-0.5-1	46.28	11	0	1899	0	14.07	4.06	281	1899
san1000	7280.00	12234	12181	4689	0.43	588.35	281.47	1126	4703
sanr200-0.9	0.75	2	0	928	0	0.18	0.08	107	928

Table 2: Results for instances of subset (ii): CPBB solves the problem already in the root node.

summarized in Table 2, whereas the results for the remaining instances are presented in Tables 3 and 4. For the search tree, Table 2/3 lists the total number of explored nodes ("nodes") and the number of nodes left ("left") after two hours of computation, the best solution value found so far ("value"), and the final gap in each of the scenarios ("gap"). In addition, Table 2/4 discusses the CPU time needed for each scenario ("time"), and in case of CPBB and BC, the total time spent for separation ("sep. time") as well as the total number of cycle inequalities that have been separated ("# ineq."). For BC, the column "new" refers to the new inequalities that are separated in addition to those in the root node of the branch-and-cut tree.

The results allow for several remarks. First of all, the tables show that a substantial performance increase could be gained by including the cycle inequalities in the root of the branch-and-cut tree. The most impressive example is the instance MANN-a81 for which almost 2 million nodes were explored by BB and a similar number still has to be explored, whereas after a single round of separation in the root node by CPBB, the integer optimal solution was found in about two seconds. In total 14 instances could not be solved within two hours with BB, but nine of them are solved by CPBB within two hours. Thereby, the number of nodes explored by CPBB is only a fraction of the number explored by BB.

Even for instances that are solved by BB within two hours, the incorporation of cycle inequalities provides significant improvements. For the 24 instances of subset (ii) that BB solved, the computation time can be reduced by 58% on average. The number of inequalities that has been separated adds up to several thousands for the larger instances. Note that these values are typically smaller than in Table 1 since Gomory cuts are generated as well. Hence, the cycle inequalities are very effective, but not found by the cut generation routines of CPLEX.

Separation of the inequalities in nodes other than the root node is less effective. Although the number of nodes needed by the branch-and-cut algorithm is reduced further, the separation is relatively time consuming, resulting in longer overall running times. The majority of the inequalities is typically separated in the root node. For those instances that cannot be solved within two hours of computing time, far less nodes are explored by BC than by CPBB. Note that for instance keller6 the exploration of the root could not be finished within two hours of computation, whereas for p-hat-1500-1 and p-hat1500-2 the computation is truncated after separation in the root is finished but before branching applied. Hence, the results of CPBB and BC do not differ for these 3 instances.

Therefore, we run all scenarios that could not be solved by BC within two hours for ten hours of CPU time. Since the root relexation of keller6 in CPBB and BC is still not solved within this period, we also run the algorithms for this instance for 24 hours. The results can be found in Table 5 and Table 6. Again, the gap is reduced significantly by inclusion of the cycle inequalities. Instance p-hat-1500-3 could be solved by CPBB in about 8 hours using a fraction of the number of nodes explored (and left) by BB. In scenario BC, separation consumes a substantial part of the CPU time. As a consequence, p-hat-1500-3 could not be solved by BC within the 10 hours.

instance	BB				CPBB				BC				opt.
	nodes	left	value	gap	nodes	left	value	gap	nodes	left	value	gap	value
c-fat500-1	45041	34533	2391	0.06	90	0	2391	0	33	0	2391	0	2391
c-fat500-2	33486	23905	2324	0.11	244	0	2324	0	139	0	2324	0	2324
c-fat500-5	28968	28946	2340	0.30	38	0	2344	0	24	0	2344	0	2344
keller6	847	837	15708	1.55	—	—	—	—	—	—	—	—	—
p-hat700-1	22090	22050	3313	0.29	222	0	3317	0	162	0	3317	0	3317
p-hat1000-1	9127	9122	4609	1.67	400	376	4637	0.40	79	74	4635	0.56	—
p-hat1000-2	13471	13461	4715	0.53	27	0	4728	0	23	0	4728	0	4728
p-hat1500-1	2644	2614	6809	2.85	—	—	6811	1.67	—	—	6811	1.67	—
p-hat1500-2	5531	5527	6892	1.51	—	—	6923	0.38	—	—	6923	0.38	—
p-hat1500-3	12400	12396	7134	0.34	59	54	7143	0.05	21	19	7142	0.09	7144

Table 3: Results for instances of subset (iii): branch-and-cut statistics.

instance	BB	CPBB			BC			
	time (sec.)	time (sec.)	sep. time (sec.)	# ineq.	time (sec.)	sep. time (sec.)	# ineq.	new
c-fat500-1	7295.20	287.60	63.55	537	416.37	207.88	715	178
c-fat500-2	7280.71	389.79	54.13	550	882.45	472.16	1091	541
c-fat500-5	7267.80	293.84	78.93	778	395.70	174.25	886	108
keller6	7391.98	7264.02	546.46	1540	7265.92	546.01	1540	0
p-hat700-1	7292.55	843.55	120.98	722	2125.75	1314.91	1216	494
p-hat1000-1	7286.09	7237.26	478.65	1758	7272.47	4489.66	2789	1031
p-hat1000-2	7283.78	1673.02	341.02	1108	2174.63	900.08	1350	242
p-hat1500-1	7296.98	7254.65	2151.03	3697	7257.63	2137.16	3697	0
p-hat1500-2	7295.38	7236.74	1272.23	3438	7236.76	1265.90	3438	0
p-hat1500-3	7285.27	7239.42	686.59	2049	7257.27	3054.47	2733	684

Table 4: Results for instances of subset (iii): CPU time and Separation statistics.

instance	BB				CPBB				BC				opt.
	nodes	left	value	gap	nodes	left	value	gap	nodes	left	value	gap	value
keller6 (10h)	7991	7981	15708	1.52	—	—	—	—	—	—	—	—	—
p-hat1000-1	52192	52145	4617	1.40	5035	4994	4640	0.27	599	575	4639	0.33	—
p-hat1500-1	21106	21076	6809	2.71	346	323	6838	1.21	99	99	6820	1.54	—
p-hat1500-2	36400	36396	6892	1.41	360	356	6923	0.29	131	122	6918	0.41	—
p-hat1500-3	67745	67741	7134	0.31	519	0	7144	0	135	53	7144	0.02	7144
keller6 (24h)	20649	20639	15708	1.51	43	43	15698	0.75	7	7	15698	0.75	—

Table 5: Results with 10/24 hours of computation: branch-and-cut statistics.

instance	BB	CPBB			BC			
	time (sec.)	time (sec.)	sep. time (sec.)	# ineq.	time (sec.)	sep. time (sec.)	# ineq.	new
keller6 (10h)	36572.81	—	—	—	—	—	—	—
p-hat1000-1	36455.69	36177.98	477.24	1758	36250.47	21730.67	4688	2930
p-hat1500-1	36420.11	36137.91	2220.73	3697	36233.13	17829.62	5037	1340
p-hat1500-2	36465.49	36137.33	1320.81	3438	36308.75	22359.38	6023	2585
p-hat1500-3	36448.44	29381.05	682.23	2049	36154.01	11746.67	4243	2194
keller6 (24h)	87972.95	86696.74	7589.31	7881	86889.86	14384.74	9602	1721

Table 6: Results with 10/24 hours of computation: CPU time and Separation statistics.

5 Concluding remarks

Stable multi-sets are a generalization of stable sets like b-matchings generalize matchings. Multi-set packings complete this analogy as generalization of set packing. In contrast to the equivalence of set packings and stable sets, stable multi-sets and multi-set packings differ fundamentally.

Although not all (cf. Section 3.1), many of the results for the stable set polytope can be imaged to the stable multi-set polytope. In this way, new insights can be gained, not only for the stable multi-set polytope but also for the well-studied stable set polytope. For example, the separation of cycle inequalities elucidates that not the length of the cycle, but in fact the bounds on the edges are the critical factor for the correctness of the algorithm. Another insight concerns chords in cycles. The result for stable multi-sets explains why chords rule out facets in the stable set case.

The polyhedral structure of the stable multi-set polytope is not only of theoretical interest, but also of computational importance. State-of-the-art integer programming solvers automatically separate clique inequalities for the binary variables in an integer program as Gomory cuts. Our computational results show that these cuts do not suffice in case of stable multi-sets. Significant improvements can be gained by the separation of cycle inequalities. Since the structure of stable multi-sets is likely to appear in more general integer programs, incorporation of this separation is worthwhile to consider.

References

[1] R. Borndörfer. *Aspects of Set Packing, Partitioning, and Covering.* PhD thesis, Technische Universität Berlin, published at Shaker Verlag Aachen, 1998.

[2] E. Cheng and S. de Vries. A note on the multi-oddity constrained shortest circuits and their separation over the stable mutli-set polytope. *Operations Research Letters*, to appear, 2003.

[3] The second DIMACS implementation challenge: $\mathcal{NP}$-Hard Problems: Maximum Clique, Graph Coloring, and Satisfiability. See http://dimacs.rutgers.edu/Challenges/ or http://mat.gsia.cmu.edu/challenge.html, 1992–1993.

[4] A. M. H. Gerards and A. Schrijver. Matrices with the Edmonds-Johnson property. *Combinatorica*, 6(4):365–379, 1986.

[5] A. M. H. Gerards and F. B. Shepherd. The graphs with all subgraphs t-perfect. *SIAM Journal on Discrete Mathematics*, 11(4):524–545, 1998.

[6] D. Gijswijt and A. Schrijver. On the b-stable set polytope of graphs without bad k_4. *SIAM Journal on Discrete Mathematics*, 16(3):511–516, 2003.

[7] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Number 2 in Algorithms and Combinatorics. Springer-Verlag, 1988.

[8] ILOG. CPLEX version 7.5, 2001. http://www.ilog.com/products/cplex.

[9] A. M. C. A. Koster and A. Zymolka. Stable Multi-Sets. *Mathematical Methods of Operations Research*, 56(1):45–65, 2002.

[10] G. L. Nemhauser and L. E. Trotter Jr. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6:48–61, 1974.

[11] M. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.

[12] A. Schrijver. *Combinatorial Optimization – Polyhedra and Efficiency.* Number 24 in Algorithms and Combinatorics. Springer-Verlag, 2003.