2012 IEEE 53rd Annual Symposium on Foundations of Computer Science

The Cutting Plane Method is Polynomial for Perfect Matchings

Karthekeyan Chandrasekaran
College of Computing
Georgia Institute of Technology
Email: karthe@gatech.edu

László A. Végh
Department of Management
London School of Economics
Email: l.vegh@lse.ac.uk

Santosh Vempala
College of Computing
Georgia Institute of Technology
Email: vempala@gatech.edu

***Abstract*—The cutting plane approach to optimal matchings has been discussed by several authors over the past decades [1]–[5], and its rate of convergence has been an open question. We prove that the cutting plane approach using Edmonds' blossom inequalities converges in polynomial time for the minimum-cost perfect matching problem. Our main insight is an LP-based method to select cutting planes. This cut selection procedure leads to a sequence of intermediate linear programs with a linear number of constraints whose optima are half-integral and supported by a disjoint union of odd cycles and edges. This structural property of the optima is instrumental in finding violated blossom inequalities (cuts) in linear time. Moreover, the number of cycles in the support of the half-integral optima acts as a potential function to show efficient convergence to an integral solution.**

***Keywords*-algorithms, matching, cutting plane methods**

I. Introduction

Integer programming is a powerful and widely used approach to modeling and solving discrete optimization problems [6], [7]. Not surprisingly, it is NP-complete and the fastest known algorithms are exponential in the number of variables (roughly $n^{O(n)}$ [8]). In spite of this intractability, integer programs of considerable sizes are routinely solved in practice. A popular approach is the cutting plane method, proposed by Dantzig, Fulkerson and Johnson [9] and pioneered by Gomory [10]–[12]. This approach can be summarized as follows:

1) Solve a linear programming relaxation (LP) of the given integer program (IP) to obtain a basic optimal solution x.
2) If x is integral, terminate. If x is not integral, find a linear inequality that is valid for the convex hull of all integer solutions but violated by x.
3) Add the inequality to the current LP, possibly drop some of the previous inequalities and solve the resulting LP to obtain a basic optimal solution x. Go back to Step 2.

For the method to be efficient, we require the following: (a) an efficient procedure for finding a violated inequality (called a cutting plane), (b) convergence of the method to an integral solution using the efficient cut-generation procedure and (c) a bound on the number of iterations to convergence. Gomory gave the first efficient cut-generation procedure and showed that the the cutting plane method implemented using his cut-generation procedure converges to an integral solution [12]. There is a rich theory on the choice of cutting planes, both in general and for specific problems of interest. This theory includes interesting families of cutting planes with efficient cut-generation procedures [10], [13]–[20], valid inequalities, closure properties and a classification of the strength of inequalities based on their *rank* with respect to cut-generating procedures [21] (e.g., the Chvátal-Gomory rank [14]), and testifies to the power and generality of the cutting plane method.

To our knowledge, however, there are no polynomial bounds on the number of iterations to convergence of the cutting plane method even for specific problems using specific cut-generation procedures. The best bound for general 0-1 integer programs remains Gomory's bound of 2^n [12]. It is possible that such a bound can be significantly improved for IPs with small Chvátal-Gomory rank [5]. A more realistic possibility is that the approach is provably efficient for combinatorial optimization problems that are known to be solvable in polynomial time. An ideal candidate could be a problem that *(a)* has a polynomial-size IP-description (the LP-relaxation is polynomial-size), and *(b)* the convex-hull of integer solutions has a polynomial-time separation oracle. Note that such problems can be solved in polynomial time via the Ellipsoid method [22]. Perhaps the first such interesting problem is minimum-cost perfect matching: *given a graph with costs on the edges, find a perfect matching of minimum total cost.*

A polyhedral characterization of the matching problem was discovered by Edmonds [23]: Basic solutions of the following linear program (extreme points of the polytope) correspond to perfect matchings of the graph.

$$\min \sum_{uv\in E} c(uv)x(uv) \qquad \text{(P)}$$
$$x(\delta(u)) = 1 \quad \forall u \in V$$
$$x(\delta(S)) \geq 1 \quad \forall S \subsetneq V, |S| \text{ odd}, 3 \leq |S| \leq |V| - 3$$
$$x \geq 0$$

A full version of this extended abstract appears on the arXiv: http://arxiv.org/abs/1207.5813. This work was supported by NSF award AF0915903. It was done while the second author was affiliated with the College of Computing, Georgia Institute of Technology, supported by NSF Grant CCF-0914732.

0272-5428/12 $26.00 © 2012 IEEE
DOI 10.1109/FOCS.2012.35

The relaxation with only the degree and nonnegativity constraints, known as the *bipartite relaxation*, suffices to characterize the convex-hull of perfect matchings in bipartite graphs, and serves as a natural starting relaxation. The inequalities corresponding to sets of odd cardinality greater than 1 are called *blossom* inequalities. These inequalities have Chvátal rank 1, i.e., applying one round of all possible Gomory cuts to the bipartite relaxation suffices to recover the perfect matching polytope of any graph [14]. Moreover, although the number of blossom inequalities is exponential in the size of the graph, for any point not in the perfect matching polytope, a violated (blossom) inequality can be found in polynomial time [1]. This suggests a natural cutting plane algorithm (Figure 1), proposed by Padberg and Rao [1] and discussed by Lovász and Plummer in their classic book on matching theory [3]. Experimental evidence suggesting that this method converges quickly was given by Grötschel and Holland [2], by Trick [4], and by Fischetti and Lodi [5]. It has been open to rigorously explain their findings. In this paper, we address the question of whether the method can be implemented to converge in polynomial time.

Figure 1. Cutting plane method for matchings

1) Start with the bipartite relaxation.
2) While the current solution is fractional,
   a) Find a violated blossom inequality and add it to the LP.
   b) Solve the new LP.

The known polynomial-time algorithms for minimum-cost perfect matching are variants of Edmonds' weighted matching algorithm [23]. It is perhaps tempting to interpret the latter as a cutting plane algorithm, by adding cuts corresponding to the shrunk sets in the iterations of Edmonds' algorithm. However, there is no correspondence between the solution x of the LP given by non-negativity and degree constraints and a family $\mathcal{F}$ of blossom inequalities, and the partial matching M in the iteration of Edmonds' algorithm when $\mathcal{F}$ is the set of shrunk nodes. In particular, the next odd set S shrunk by Edmonds' algorithm might not even be a cut for x (i.e., $x(\delta(S)) \geq 1$). It is even possible, that the bipartite relaxation already has an integer optimal solution, whereas Edmonds' algorithm proceeds by shrinking and unshrinking a long sequence of odd sets.

The bipartite relaxation has the nice property that any basic solution is half-integral and its support is a disjoint union of edges and odd cycles. This makes it particularly easy to find violated blossom inequalities – any odd component of the support gives one. This is also the simplest heuristic that is employed in the implementations [2], [4] for finding violated blossom inequalities. However, if we have a fractional solution in a later phase, there is no guarantee that we can find an odd connected component whose blossom inequality is violated, and therefore sophisticated and significantly slower separation methods are needed for finding cutting planes, e.g., the Padberg-Rao procedure [1]. Thus, it is natural to wonder if there is a choice of cutting planes that maintains half-integrality of intermediate LP optimal solutions.

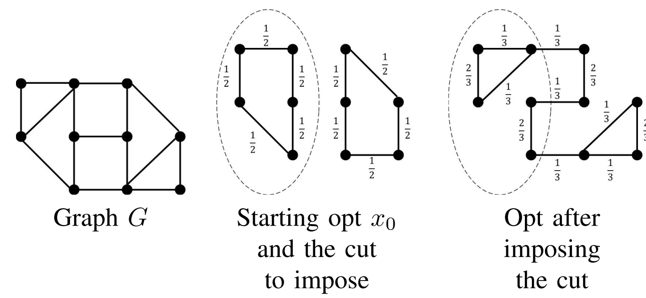

Figure 2. Counterexample to the half-integrality conjecture. All edge costs are one.

At first sight, maintaining half-integrality seems to be impossible. Figure 2 shows an example where the starting solution consists of two odd cycles. There is only one reasonable way to impose cuts, and it leads to a non half-integral basic feasible solution. Observe however, that in the example, the bipartite relaxation also has an integer optimal solution. The problem here is the existence of multiple basic optimal solutions. To avoid such degeneracy, we will ensure that all linear systems that we encounter have unique optimal solutions.

This uniqueness is achieved by a simple deterministic perturbation of the integer cost function, which increases the input size polynomially. We observe that this perturbation is only a first step towards maintaining half-integrality of intermediate LP optima. More careful cut retention and cut addition procedures are needed to maintain half-integrality.

A. Main result

We call a vector $x \in \mathbb{R}^E$ *proper-half-integral* if $x(e) \in \{0, 1/2, 1\}$ for every $e \in E$ and $\text{supp}(x)$ is a disjoint union of edges and odd cycles. It is well-known that every basic feasible soultion to the bipartite relaxation has this property.

A family $\mathcal{F}$ of subsets of V is called *laminar*, if for any $X, Y \in \mathcal{F}$, one of $X \cap Y = \emptyset$, $X \subseteq Y$, $Y \subseteq X$ holds. Given an integer cost function $c : E \to \mathbb{Z}$ on the edges of a graph $G = (V, E)$, let us define the *perturbation* $\tilde{c}$ by ordering the edges arbitrarily, and increasing the cost of edge i by $1/2^i$.

We are now ready to state our main theorem.

Theorem 1. *Let $G = (V, E)$ be a graph on n nodes with edge costs $c : E \to \mathbb{Z}$ and let $\tilde{c}$ denote the perturbation of c. Then, there exists an implementation of the cutting plane method that finds the minimum $\tilde{c}$-cost perfect matching such that*

(i) every intermediate LP is defined by the bipartite relaxation constraints and a collection of blossom inequalities corresponding to a laminar family of odd subsets,
(ii) every intermediate LP optimum is unique and proper-half-integral, and
(iii) the total number of iterations to arrive at a minimum $\tilde{c}$-cost perfect matching is $O(n \log n)$.

Moreover, the collection of blossom inequalities used at each step can be identified by solving an LP of the same size as the current LP. The minimum $\tilde{c}$-cost perfect matching is also a minimum c-cost perfect matching.

To our knowledge, this is the first polynomial bound on the convergence of a cutting plane method for matchings. It is easy to verify that for an n-vertex graph, a laminar family of nontrivial odd sets may have at most $n/2$ members, hence every intermediate LP has at most $3n/2$ inequalities apart from the non-negativity constraints.

B. Cut selection via dual values

Ensuring unique optimal solutions itself does not suffice to maintain proper-half-integrality of optimal solutions upon adding any sequence of blossom inequalities. This is the case even if using laminar families of blossom inequalities. Thus a careful choice is to be made while selecting new cuts and it is also crucial that we eliminate certain older ones.

At any iteration, inequalities that are tight for the current optimal solution are natural candidates for retaining in the next iteration while the new inequalities are determined by odd cycles in the support of the current optimal solution. However, it turns out that keeping all tight inequalities does not maintain half-integrality. Our main algorithmic insight is that the choice of cuts for the next iteration can be determined by examining optimal dual solutions to the current LP – we retain those cuts whose dual values are strictly positive. Since there could be multiple dual optimal solutions, we use a restricted type of dual optimal solution (later called *positively-critical dual* in this paper) that can be computed either by solving a single LP of the same complexity or combinatorially. Moreover, we also ensure that the set of cuts imposed in any LP are laminar and correspond to blossom inequalities.

Eliminating cutting planes that have zero dual values in any later iteration is common in most implementations of the cutting plane algorithm; although this is done mainly to keep the number of inequalities from blowing up, another justification is that a cut with zero dual value is not a facet contributing to the current LP optimum.

C. Algorithm C-P-Matching

Let $G = (V, E)$ be a graph, $c : E \to \mathbb{R}$ a cost function on the edges, and assume G has a perfect matching. Without loss of generality we may assume that $c \geq 0$.

Let $\mathcal{O}$ be the set of all odd subsets of V of size at least 3, and let $\mathcal{V}$ denote the set of one element subsets of V. For a family of odd sets $\mathcal{F} \subseteq \mathcal{O}$, consider the following pair of linear programs.

$$\begin{array}{rl}
P_{\mathcal{F}}(G,c) & D_{\mathcal{F}}(G,c) \\
\min \sum_{uv \in E} c(uv)x(uv) & \max \sum_{S \in \mathcal{V} \cup \mathcal{F}} \Pi(S) \\
x(\delta(u)) = 1 \forall u \in \mathcal{V} & \sum_{S \in \mathcal{V} \cup \mathcal{F} : uv \in \delta(S)} \Pi(S) \leq c(uv) \\
 & \forall uv \in E \\
x \geq 0 & \Pi(S) \geq 0 \forall S \in \mathcal{F}
\end{array}$$

$\mathcal{F} = \emptyset$ gives the bipartite relaxation, also denoted by $P_0(G,c)$ and $D_0(G,c)$. For $\mathcal{F} = \mathcal{O}$, the formulation is identical to (P). Every intermediate LP in our cutting plane algorithm will be $P_{\mathcal{F}}(G,c)$ for some laminar family $\mathcal{F}$. We will use $\Pi(v)$ to denote $\Pi(\{v\})$ for dual solutions.

Assume we are given a dual feasible solution Γ to $D_{\mathcal{F}}(G,c)$. We say that a dual optimal solution Π to $D_{\mathcal{F}}(G,c)$ is Γ-*extremal*, if it minimizes

$$h(\Pi, \Gamma) = \sum_{S \in \mathcal{V} \cup \mathcal{F}} \frac{|\Pi(S) - \Gamma(S)|}{|S|}$$

among all dual optimal solutions Π. A Γ-extremal dual optimal solution can be found by solving a single LP if we are provided with the primal optimal solution to $P_{\mathcal{F}}(G,c)$ (see Section V-B).

Figure 3. Algorithm C-P-Matching

1) Let c denote the cost function on edges after perturbation (i.e., after ordering the edges arbitrarily and increasing the cost of edge i by $1/2^i$).
2) **Starting LP.** Let $\mathcal{F} = \emptyset$. The starting LP, $P_{\mathcal{F}}(G,c)$, is the bipartite relaxation and the starting dual Γ is identically zero.
3) **Repeat** until x is integral:
   a) **Solve LP.** Find an optimal solution x to $P_{\mathcal{F}}(G,c)$.
   b) **Choose old cutting planes.** Find a Γ-extremal dual optimal solution Π to $D_{\mathcal{F}}(G,c)$. Let
   $$\mathcal{H}' = \{S \in \mathcal{F} : \Pi(S) > 0\}.$$
   c) **Choose new cutting planes.** Let $\mathcal{C}$ denote the set of odd cycles in supp(x). For each $C \in \mathcal{C}$, define $\hat{C}$ as the union of $V(C)$ and the maximal sets of $\mathcal{H}'$ intersecting it. Let
   $$\mathcal{H}'' = \{\hat{C} : C \in \mathcal{C}\}.$$
   d) Set the next $\mathcal{F} = \mathcal{H}' \cup \mathcal{H}''$ and $\Gamma = \Pi$.
4) **Return** the minimum-cost perfect matching x.

The proposed cutting plane implementation is shown in Figure 3. From the previous set of cuts, we retain only those which have a positive value in an extremal dual optimal solution; let $\mathcal{H}'$ denote this set of cuts. The new set of cuts $\mathcal{H}''$ correspond to odd cycles in the support of the current solution. However, in order to maintain laminarity of the cut family, we do not add the vertex sets of these cycles but instead their union with all the sets in $\mathcal{H}'$ that they intersect. We will show that these unions are also odd sets and thus give blossom inequalities. In the first iteration, there is no need to solve the dual LP as $\mathcal{F}$ will be empty.

II. Factor-critical sets

The notion of factor-critical sets and factor-critical duals play a central role in showing half-integrality of intermediate solutions. These are extensions of concepts central to the analysis of Edmonds' algorithm.

Let $H = (V, E)$ be a graph and $\mathcal{F}$ be a laminar family of subsets of V. We say that an edge set $M \subseteq E$ is an *$\mathcal{F}$-matching*, if it is a matching, and for any $S \in \mathcal{F}$, $|M \cap \delta(S)| \leq 1$. For a set $S \subseteq V$, we call a set M of edges to be an *$(S, \mathcal{F})$-perfect-matching* if it is an $\mathcal{F}$-matching covering precisely the vertex set S.

A set $S \in \mathcal{F}$ is defined to be *$(H, \mathcal{F})$-factor-critical* or *$\mathcal{F}$-factor-critical in H*, if for every node $u \in S$, there exists an $(S \setminus \{u\}, \mathcal{F})$-perfect-matching using the edges of H. For a laminar family $\mathcal{F}$ and a feasible solution Π to $D_{\mathcal{F}}(G, c)$, let $G_\Pi = (V, E_\Pi)$ denote the graph of tight edges. For simplicity we will say that a set $S \in \mathcal{F}$ is $(\Pi, \mathcal{F})$-factor-critical if it is $(G_\Pi, \mathcal{F})$-factor critical, i.e., S is $\mathcal{F}$-factor-critical in G_Π. For a vertex $u \in S$, corresponding matching M_u is called the *Π-critical-matching* for u. If $\mathcal{F}$ is clear from the context, then we simply say S is Π-factor-critical.

A feasible solution Π to $D_{\mathcal{F}}(G, c)$ is an *$\mathcal{F}$-critical* dual, if every $S \in \mathcal{F}$ is $(\Pi, \mathcal{F})$-factor-critical, and $\Pi(T) > 0$ for every non-maximal set T of $\mathcal{F}$. A family $\mathcal{F} \subseteq \mathcal{O}$ is called a *critical family*, if $\mathcal{F}$ is laminar, and there exists an $\mathcal{F}$-critical dual solution. This will be a crucial notion: the set of cuts imposed in every iteration of the cutting plane algorithm will be a critical family. The following observation provides some context and motivation for these definitions.

Proposition 2. *Let $\mathcal{F}$ be the set of contracted sets at some stage of Edmonds' matching algorithm. Then the corresponding dual solution Π in the algorithm is an $\mathcal{F}$-critical dual.* □

We call Π to be an *$\mathcal{F}$-positively-critical* dual, if Π is a feasible solution to $D_{\mathcal{F}}(G, c)$, and every $S \in \mathcal{F}$ such that $\Pi(S) > 0$ is $(\Pi, \mathcal{F})$-factor-critical. Clearly, every $\mathcal{F}$-critical dual is also an $\mathcal{F}$-positively-critical dual, but the converse is not true. The extremal dual optimal solutions found in every iteration of Algorithm C-P-Matching will be $\mathcal{F}$-positively-critical, where $\mathcal{F}$ is the family of blossom inequalities imposed in that iteration.

The following *uniqueness* property is used to guarantee the existence of a proper-half-integral solution in each step. We require that the cost function $c : E \to \mathbb{R}$ satisfies:

For every critical family $\mathcal{F}$, $P_{\mathcal{F}}(G, c)$ has a unique optimal solution. *(*)*

Lemma 3. *Let $c : E \to \mathbb{Z}$ be an integer cost function, and $\tilde{c}$ be its perturbation. Then $\tilde{c}$ satisfies the uniqueness property (*).*

III. Analysis outline and proof of the main theorem

The proof of our main theorem is established in two parts. In the first part, we show that half-integrality of the intermediate primal optimum solutions is guaranteed by the existence of an $\mathcal{F}$-positively-critical dual optimal solution to $D_{\mathcal{F}}(G, c)$.

Lemma 4. *Let $\mathcal{F}$ be a laminar odd family and assume $P_{\mathcal{F}}(G, c)$ has a unique optimal solution x. If there exists an $\mathcal{F}$-positively-critical dual optimal solution, then x is proper-half-integral.*

The proof is outlined in Section IV using contraction techniques. The next lemma shows that if $\mathcal{F}$ is a critical family, then the extremal dual optimal solutions found in the algorithm are in fact $\mathcal{F}$-positively-critical dual optimal solutions.

Lemma 5. *Suppose that in an iteration of Algorithm C-P-Matching, $\mathcal{F}$ is a critical family with Γ being an $\mathcal{F}$-critical dual solution. Then a Γ-extremal dual optimal solution Π is an $\mathcal{F}$-positively-critical dual optimal solution. Moreover, the next set of cuts $\mathcal{H} = \mathcal{H}' \cup \mathcal{H}''$ is a critical family with Π being an $\mathcal{H}$-critical dual.*

The proof sketch in Section V is based on analyzing the structure of dual optimal solutions. Lemmas 4 and 5 together guarantee that the unique primal optimal solutions obtained during the execution of the algorithm are proper-half-integral. In the second part of the proof of the main theorem, we show convergence by considering the number of odd cycles, $\text{odd}(x)$, in the support of the current primal optimal solution x.

Lemma 6. *Assume the cost function c satisfies (*). Then $odd(x)$ is non-increasing during the execution of Algorithm C-P-Matching.*

We observe that similar to Lemma 4, the above Lemma 6 is also true if we choose an arbitrary $\mathcal{F}$-positively-critical dual optimal solution Π in each iteration of the algorithm. To show that the number of cycles cannot remain the same and has to strictly decrease within a polynomial number of iterations, we need the more specific choice of extremal duals.

Lemma 7. *Assume the cost function c satisfies (*) and that* $\text{odd}(x)$ *does not decrease between iterations* i *and* j*, for some* $i < j$*. Let* $\mathcal{F}_k$ *be the set of blossom inequalities imposed in the* k*'th iteration and* $\mathcal{H}''_k = \mathcal{F}_k \setminus \mathcal{F}_{k-1}$ *be the subset of new inequalities in this iteration. Then,*

$$\bigcup_{k=i+1}^{j} \mathcal{H}''_k \subseteq \mathcal{F}_{j+1}.$$

We prove this progress by coupling intermediate primal and dual solutions with the solutions of a *Half-integral Matching* algorithm, a variation of Edmonds' primal-dual weighted matching algorithm [23] that we design for this purpose. Our argument needs one phase of this algorithm and this is what we analyze in detail in Section VI-A. Although we introduce it for analysis, we note that the algorithm can be extended to a strongly-polynomial combinatorial algorithm for minimum-cost perfect matching. Unlike Edmonds' algorithm, which maintains an integral matching and extends the matching to cover all vertices, this extended algorithm maintains a proper half-integral solution in every iteration.

The half-integral matching algorithm starts from a partial matching x in G, leaving a set W of nodes exposed, and a dual Π whose support is a laminar family $\mathcal{V} \cup \mathcal{F}$ with $\mathcal{F} \subseteq \mathcal{O}$; x and Π satisfy primal-dual slackness conditions. The algorithm transforms x to a proper-half-integral perfect matching and Π to a dual solution with support contained in $\mathcal{V} \cup \mathcal{F}$, satisfying complementary slackness. We now give a sketch of the proofs of Lemmas 6 and 7 using the algorithm.

Let us consider two consecutive primal solutions x_i and x_{i+1} in the cutting plane algorithm, with duals Π_i and Π_{i+1}. We contract every set $S \in \mathcal{O}$ with $\Pi_{i+1}(S) > 0$; let $\hat{G}$ be the resulting graph. By Lemma 4 the image x'_{i+1} of x_{i+1} is the unique optimal solution to the bipartite relaxation in $\hat{G}$. The image x'_i of x_i is proper-half-integral in $\hat{G}$ with some exposed nodes W; let Π'_i be the image of Π_i. Every exposed node in W corresponds to a cycle in $\text{supp}(x_i)$. We start in $\hat{G}$ with the solutions x'_i and Π'_i, and we prove that it must terminate with the primal solution x'_{i+1}. The analysis of the half-integral matching algorithm reveals that the total number of exposed nodes and odd cycles does not increase; this will imply Lemma 6.

To prove Lemma 7, we show that if the number of cycles does not decrease between phases i and $i+1$, then the algorithm also terminates with the extremal dual optimal solution Π'_{i+1}. This enables us to couple the performance of Half-integral Matching between phases i and $i+1$ and between $i+1$ and $i+2$: the (alternating forest) structure built in the former iteration carries over to the latter one. As a consequence, all cuts added in iteration i will be imposed in all subsequent phases until the number of odd cycles decreases.

Proof of Theorem 1: We use Algorithm C-P-Matching given in Figure 3 for a perturbed cost function. By Lemma 3, this satisfies (*). Let i denote the index of the iteration. We prove by induction on i that every intermediate solution x_i is proper-half-integral and (i) follows immediately by the choice of the algorithm. The initial solution x_0 is clearly proper-half-integral. The induction step follows by Lemmas 4 and 5 and the uniqueness property. Further, by Lemma 6, the number of odd cycles in the support does not increase.

Assume the number of cycles in the i'th phase is ℓ, and we have the same number of odd cycles ℓ in a later iteration j. Between iterations i and j, the set $\mathcal{H}''_k$ always contains ℓ cuts, and thus the number of cuts added is at least $\ell(j-i)$. By Lemma 7, all cuts in $\bigcup_{k=i+1}^{j} \mathcal{H}''_k$ are imposed in the family $\mathcal{F}_{j+1}$. Since $\mathcal{F}_{j+1}$ is a laminar odd family, it can contain at most $n/2$ subsets, and therefore $j - i \leq n/2\ell$. Consequently, the number of cycles must decrease from ℓ to $\ell - 1$ within $n/2\ell$ iterations. Since $\text{odd}(x_0) \leq n/3$, the number of iterations is at most $O(n \log n)$. Finally, it is easy to verify that the optimal solution for $\tilde{c}$ is also optimal for c. ■

IV. Contractions and half-integrality

In this section, we introduce contractions to prove Lemma 4. Given Π, an $\mathcal{F}$-positively-critical dual optimal solution for the laminar odd family $\mathcal{F}$, we show (Lemma 8) that contracting every set $S \in \mathcal{F}$ with $\Pi(S) > 0$ preserves primal and dual optimal solutions (similar to Edmonds' primal-dual algorithm). Moreover, if we had a unique primal optimal solution x to $P_{\mathcal{F}}(G, c)$, its image x' in the contracted graph is the unique optimal solution; if x' is proper-half-integral, then so is x.

Let $\mathcal{F}$ be a laminar odd family, Π be a feasible solution to $D_{\mathcal{F}}(G, c)$, and let $S \in \mathcal{F}$ be a $(\Pi, \mathcal{F})$-factor-critical set. Let us define

$$\Pi_S(u) := \sum_{T \in \mathcal{V} \cup \mathcal{F}: T \subsetneq S, u \in T} \Pi(T)$$

to be the total dual contribution of sets inside S containing u.

By contracting S w.r.t. Π, we mean the following: Let $G' = (V', E')$ be the contracted graph on node set $V' = (V \setminus S) \cup \{s\}$, s representing the contraction of S. Let $\mathcal{V}'$ denote the set of one-element subsets of V'. For a set $T \subseteq V$, let T' denote its contracted image. Let $\mathcal{F}'$ be the set of nonsingular images of the sets of $\mathcal{F}$, that is, $T' \in \mathcal{F}'$ if $T \in \mathcal{F}$, and $T' \setminus \{s\} \neq \emptyset$. Let E' contain all edges $uv \in E$ with $u, v \notin S$ and for every edge uv with $u \in S$, $v \in V - S$ add an edge sv. Let us define the image Π' of Π to be $\Pi'(T') = \Pi(T)$ for every $T' \in \mathcal{V}' \cup \mathcal{F}'$ and the image x' of x to be $x'(u'v') = x(uv)$. Define the new edge costs

$$c'(u'v') = \begin{cases} c(uv) & \text{if } uv \in E[V \setminus S], \\ c(uv) - \Pi_S(u) & \text{if } u \in S,\ v \in V \setminus S. \end{cases}$$

Lemma 8. *Let $\mathcal{F}$ be a laminar odd family, x be an optimal solution to $P_{\mathcal{F}}(G,c)$, Π be a feasible solution to $D_{\mathcal{F}}(G,c)$. Let $S \in \mathcal{F}$ be a $(\Pi,\mathcal{F})$-factor-critical set, and let $G', c', \mathcal{F}'$ denote the graph, costs and laminar family respectively obtained by contracting S w.r.t. Π and let x', Π' be the images of x, Π respectively. Then the following hold.*

(i) Π' is a feasible solution to $D_{\mathcal{F}'}(G',c')$. Furthermore, if a set $T \in \mathcal{F}$, $T \setminus S \neq \emptyset$ is $(\Pi,\mathcal{F})$-factor-critical, then its image T' is $(\Pi',\mathcal{F}')$-factor-critical.

(ii) Suppose Π is an optimal solution to $D_{\mathcal{F}}(G,c)$ and $x(\delta(S)) = 1$. Then x' is an optimal solution to $P_{\mathcal{F}'}(G',c')$ and Π' is optimal to $D_{\mathcal{F}'}(G',c')$.

(iii) If x is the unique optimum to $P_{\mathcal{F}}(G,c)$, and Π is an optimal solution to $D_{\mathcal{F}}(G,c)$, then x' is the unique optimum to $P_{\mathcal{F}'}(G',c')$. Moreover, x' is proper-half-integral if and only if x is proper-half-integral. Further, assume C' is an odd cycle in supp(x') and let T be the pre-image of $V(C')$ in G. Then, supp(x) inside T consists of an odd cycle and matching edges.

Corollary 9. *Assume x is the optimal solution to $P_{\mathcal{F}}(G,c)$ and there exists an $\mathcal{F}$-positively-critical dual optimum Π. Let $\hat{G}, \hat{c}$ be the graph, and cost obtained by contracting all maximal sets $S \in \mathcal{F}$ with $\Pi(S) > 0$ w.r.t. Π, and let $\hat{x}$ be the image of x in $\hat{G}$.*

(i) $\hat{x}$ and $\hat{\Pi}$ are the optimal solutions to the bipartite relaxation $P_0(\hat{G},\hat{c})$ and $D_0(\hat{G},\hat{c})$ respectively.

(ii) If x is the unique optimum to $P_{\mathcal{F}}(G,c)$, then $\hat{x}$ is the unique optimum to $P_0(\hat{G},\hat{c})$. If $\hat{x}$ is proper-half-integral, then x is also proper-half-integral.

Proof of Lemma 4: Let Π be an $\mathcal{F}$-positively-critical dual optimum, and let x be the unique optimal solution to $P_{\mathcal{F}}(G,c)$. Contract all maximal sets $S \in \mathcal{F}$ with $\Pi(S) > 0$, obtaining the graph $\hat{G}$ and cost $\hat{c}$. Let $\hat{x}$ be the image of x in $\hat{G}$. By Corollary 9(ii), $\hat{x}$ is unique optimum to the bipartite relaxation $P_0(\hat{G},\hat{c})$. Consequently, $\hat{x}$ is proper-half-integral and hence by Corollary 9(ii), x is also proper-half-integral. ■

V. Structure of dual solutions

In this section, we show two properties about positively-critical dual optimal solutions – (1) an optimum Ψ to $D_{\mathcal{F}}(G,c)$ can be transformed into an $\mathcal{F}$-positively-critical dual optimum (Section V-A) if $\mathcal{F}$ is a critical family and (2) a Γ-extremal dual optimal solution to $D_{\mathcal{F}}(G,c)$ as obtained in the algorithm is also an $\mathcal{F}$-positively-critical dual optimal solution (Section V-B).

Assume $\mathcal{F} \subseteq \mathcal{O}$ is a critical family, with Π being an $\mathcal{F}$-critical dual solution, and let Ψ be an arbitrary dual optimal solution to $D_{\mathcal{F}}(G,c)$. Consider a set $S \in \mathcal{F}$. We say that the dual solutions Π and Ψ are *identical* inside S, if $\Pi(T) = \Psi(T)$ for every set $T \subsetneq S$, $T \in \mathcal{F} \cup \mathcal{V}$. We defined $\Pi_S(u)$ in the previous section; we also use the analogous notation for Ψ. Let us now define

$$\Delta_{\Pi,\Psi}(S) := \max_{u \in S}\left(\Pi_S(u) - \Psi_S(u)\right).$$

We say that Ψ is *consistent* with Π inside S, if $\Pi_S(u) - \Psi_S(u) = \Delta_{\Pi,\Psi}(S)$ holds for every $u \in S$ that is incident to an edge $uv \in \delta(S) \cap \text{supp}(x)$. We derive the following important structural lemmas.

Lemma 10. *Let $\mathcal{F} \subseteq \mathcal{O}$ be a critical family, with Π being an $\mathcal{F}$-critical dual solution and let Ψ be an optimal solution to $D_{\mathcal{F}}(G,c)$. Let x be an optimal solution to $P_{\mathcal{F}}(G,c)$. Then Ψ is consistent with Π inside every set $S \in \mathcal{F}$ such that $x(\delta(S)) = 1$.*

Lemma 11. *Given a laminar odd family $\mathcal{F} \subset \mathcal{O}$, let Λ and Γ be two dual feasible solutions to $D_{\mathcal{F}}(G,c)$. If a subset $S \in \mathcal{F}$ is both $(\Lambda,\mathcal{F})$-factor-critical and $(\Gamma,\mathcal{F})$-factor-critical, then Λ and Γ are identical inside S.*

A. Finding a positively-critical dual optimal solution

Let $\mathcal{F} \subseteq \mathcal{O}$ be a critical family with Π being an $\mathcal{F}$-critical dual. Let Ψ be a dual optimum solution to $D_{\mathcal{F}}(G,c)$. Our goal is to satisfy the property that for every $S \in F$, if $\Psi(S) > 0$, then Ψ and Π are identical inside S. By Lemma 11, it is equivalent to showing that Ψ is $\mathcal{F}$-positively-critical. We modify Ψ by the algorithm shown in Figure 4. The correctness of the algorithm follows by showing that the modified solution $\bar{\Psi}$ is also dual optimal, and it is closer to Π.

Figure 4. Algorithm Positively-critical-dual-opt

1) **Repeat** while Ψ is not $\mathcal{F}$-positively-critical dual.
   a) Choose a maximal set $S \in \mathcal{F}$ with $\Psi(S) > 0$, such that Π and Ψ are not identical inside S.
   b) Set $\Delta := \Delta_{\Pi,\Psi}(S)$.
   c) Let $\lambda := \min\{1, \Psi(S)/\Delta\}$ if $\Delta > 0$ and $\lambda := 1$ if $\Delta = 0$.
   d) Replace Ψ by the following $\bar{\Psi}$.
   $$\bar{\Psi}(T) \quad := \begin{cases} (1-\lambda)\Psi(T) + \lambda\Pi(T) \text{ if } T \subsetneq S, \\ \Psi(S) - \Delta\lambda \text{ if } T = S, \\ \Psi(T) \text{ otherwise }. \end{cases}$$
2) **Return** $\mathcal{F}$-positively-critical dual optimum Ψ.

Lemma 12. *Let $\mathcal{F}$ be a critical family with Π being an $\mathcal{F}$-critical dual feasible solution. Algorithm Positively-critical-dual-opt in Figure 4 transforms an arbitrary dual optimal solution Ψ to an $\mathcal{F}$-positively-critical dual optimal solution in at most $|\mathcal{F}|$ iterations.*

B. Extremal dual solutions

In this section, we prove Lemma 5. Assume $\mathcal{F} \subseteq \mathcal{O}$ is a critical family, with Π being an $\mathcal{F}$-critical dual. Let x be the unique optimal solution to $P_{\mathcal{F}}(G,c)$. Let $\mathcal{F}_x = \{S \in \mathcal{F} : x(\delta(S)) = 1\}$ the collection of tight sets for x. A Π-extremal dual can be found by solving the following LP.

$$\min h(\Psi,\Pi) = \sum_{S\in\mathcal{V}\cup\mathcal{F}_x} \frac{r(S)}{|S|} \qquad (D^*_{\mathcal{F}})$$
$$-r(S) \le \Psi(S) - \Pi(S) \le r(S) \quad \forall S \in \mathcal{V}\cup\mathcal{F}_x$$
$$\sum_{S\in\mathcal{V}\cup\mathcal{F}_x : uv\in\delta(S)} \Psi(S) = c(uv) \quad \forall uv \in \text{supp}(x)$$
$$\sum_{S\in\mathcal{V}\cup\mathcal{F}_x : uv\in\delta(S)} \Psi(S) \le c(uv) \quad \forall uv \in E\setminus\text{supp}(x)$$
$$\Psi(S) \ge 0 \quad \forall S \in \mathcal{F}_x$$

The support of Ψ is restricted to sets in $\mathcal{V}\cup\mathcal{F}_x$. Primal-dual slackness implies that the feasible solutions to this program coincide with the optimal solutions of $D_{\mathcal{F}}(G,c)$, hence an optimal solution to $D^*_{\mathcal{F}}$ is also an optimal solution to $D_{\mathcal{F}}(G,c)$.

Lemma 13. *Let $\mathcal{F}\subset\mathcal{O}$ be a critical family with Π being an $\mathcal{F}$-critical dual. Then, a Π-extremal dual is also an $\mathcal{F}$-positively-critical dual optimal solution.*

The proof of Lemma 5 follows using Lemmas 4 and 13.

VI. Convergence

The structural properties given in Lemmas 6 and 7 are established as follows. First, we give a variant of Edmonds' primal-dual algorithm for half-integral matchings and show that it converges quickly. Next, we argue that applying this algorithm to the current primal/dual solution leads to an optimal solution of the next LP. As a consequence we get that the number of odd cycles in the support of the primal is nonincreasing. Finally, we prove the extremal dual solution of the next LP must be *identical* to the one found by this combinatorial algorithm. From the structure of the dual solution, we can infer that cuts that are added in a sequence of iterations where the number of odd cycles does not decrease are not dropped at any point and therefore cannot be too many (since they also form a laminar family).

A. The half-integral matching algorithm

The algorithm will be applied in certain contractions of G, but here we present it for a general graph $G=(V,E)$ and cost c. We use the terminology of Edmonds' weighted matching algorithm [23] as described by Schrijver [24, Vol A, Chapter 26].

Let $W\subseteq V$, and let $\mathcal{F}\subset\mathcal{O}$ be a laminar family of odd sets that are disjoint from W. Let $\mathcal{V}^W$ denote the set of one-element subsets of $V\setminus W$. The following primal $P^W_{\mathcal{F}}(G,c)$ and dual $D^W_{\mathcal{F}}(G,c)$ programs describe fractional matchings that leave the set of nodes in W exposed (unmatched) while satisfying the blossom inequalities corresponding to a laminar family $\mathcal{F}$. The primal program is identical to $P_{\mathcal{F}}(G\setminus W,c)$ while optimal solutions to $D_{\mathcal{F}}(G\setminus W,c)$ that are feasible to $D^W_{\mathcal{F}}(G,c)$ are also optimal solutions to $D^W_{\mathcal{F}}(G,c)$.

$P^W_{\mathcal{F}}(G,c)$	$D^W_{\mathcal{F}}(G,c)$
$\min \sum_{uv\in E} c(uv)x(uv)$	$\max \sum_{S\in\mathcal{V}^W\cup\mathcal{F}} \Pi(S)$
$x(\delta(u)) = 1 \forall u\in V-W$	$\sum_{S\in\mathcal{V}^W\cup\mathcal{F}: uv\in\delta(S)} \Pi(S) \le c(uv) \quad \forall uv\in E$
$x(\delta(u)) = 0 \forall u\in W$	$\Pi(S)\ge 0 \forall S\in\mathcal{F}$
$x(\delta(S)) \ge 1 \forall S\in\mathcal{F}$	

The algorithm is iterative. In each iteration, it maintains a set $T\subseteq W$, a subset $\mathcal{L}\subseteq\mathcal{F}$ of cuts, a proper-half-integral optimal solution z to $P^T_{\mathcal{L}}(G,c)$, and an $\mathcal{L}$-critical dual optimal solution Λ to $D^T_{\mathcal{L}}(G,c)$ such that $\Lambda(S)>0$ for every $S\in\mathcal{L}$. In the beginning $T=W$, $\mathcal{L}=\mathcal{F}$ and the algorithm terminates when $T=\emptyset$.

We work on the graph $G^*=(\mathcal{V}^*,E^*)$, obtained the following way from G: We first remove every edge in E that is not tight w.r.t. Λ, and then contract all maximal sets of $\mathcal{L}$ w.r.t. Λ. The node set of $\mathcal{V}^*$ is identified with the pre-images. Let c^* denote the contracted cost function and z^* the image of z. Since E^* consists only of tight edges, $\Lambda(u)+\Lambda(v)=c^*(uv)$ for every edge $uv\in E^*$. Since $\mathcal{F}$ is disjoint from W, the nodes in $\mathcal{L}$ will always have degree 1 in z^*.

In the course of the algorithm, we may decrease $\Lambda(S)$ to 0 for a maximal set S of $\mathcal{L}$. In this case, we remove S from $\mathcal{L}$ and modify G^*, c^* and z^* accordingly. This operation will be referred as 'unshrinking' S. New sets will never be added to $\mathcal{L}$.

The algorithm works by modifying the solution z^* and the dual solution Λ^*. An edge $uv\in E^*$ is called a 0-edge/$\frac{1}{2}$-edge/1-edge according to the value $z^*(uv)$. A modification of z^* in G^* can be naturally extended using Λ-critical-matchings inside S.

A walk $P=v_0v_1v_2\dots v_k$ in G^* is called an alternating walk, if every odd edge is a 0-edge and every even edge is a 1-edge. If every node occurs in P at most once, it is called an alternating path. By *alternating along the path P*, we mean modifying $z^*(v_iv_{i+1})$ to $1-z^*(v_iv_{i+1})$ on every edge of P. If k is odd, $v_0=v_k$ and no other node occurs twice, then P is called a *blossom* with base v_0.

Claim 14 ([24, Thm 24.3]). *Let $P=v_0v_1\dots v_{2k+1}$ be an alternating walk. Either P is an alternating path, or it contains a blossom C and an even alternating path from v_0 to the base of the blossom.*

The algorithm is described in the above figure. The scenarios in *Case I* are illustrated in Figure 5. In Case II, we

Half-integral Matching

Input. A subset $W \subseteq V$, a critical family $\mathcal{F} \subset \mathcal{O}$ with all sets in $\mathcal{F}$ disjoint from W, a proper-half-integral optimal solution w to $P_{\mathcal{F}}^{W}(G,c)$, and an $\mathcal{F}$-critical dual optimal solution Γ to $D_{\mathcal{F}}^{W}(G,c)$.

Output. A proper-half-integral optimal solution z to $P_{\mathcal{L}}(G,c)$ and an $\mathcal{L}$-critical dual optimal solution Λ to $P_{\mathcal{L}}(G,c)$ for some $\mathcal{L} \subseteq \mathcal{F}$.

1) Initialize $z = w$, $\mathcal{L} = \mathcal{F}$, $\Lambda = \Gamma$, and $T = W$. Let $G^* = (\mathcal{V}^*, E^*)$, where $E^* \subseteq E$ are edges that are tight w.r.t. Λ, and all maximal sets of $\mathcal{L}$ w.r.t. Λ are contracted; c^* and z^* are defined by the contraction. Let $R \supseteq T$ be the set of exposed nodes and nodes incident to $\frac{1}{2}$-edges in z^*.
2) While T is not empty,
   Case I: There exists an alternating T-R-walk in G^*. Let $P = v_0 \dots v_{2k+1}$ denote a shortest such walk.
   (a) If P is an alternating path, and $v_{2k+1} \in T$, then change z by alternating along P.
   (b) If P is an alternating path, and $v_{2k+1} \in R - T$, then let C denote the odd cycle containing v_{2k+1}. Change z by alternating along P, and replacing z on C by a blossom with base v_{2k+1}.
   (c) If P is not a path, then by Claim 14, it contains an even alternating path P_1 to a blossom C. Change z by alternating along P_1, and setting $z^*(uv) = 1/2$ on every edge of C.
   Case II: There exists no alternating T-R-walk in G^*. Define $\mathcal{B}^+ := \{S \in \mathcal{V}^* : \exists$an even alternating path from T to $S\}$, $\mathcal{B}^- := \{S \in \mathcal{V}^* : \exists$an odd alternating path from T to $S\}$.
   For some $\varepsilon > 0$, reset
$$\Lambda(S) := \begin{cases} \Lambda(S) + \varepsilon & \text{if } S \in \mathcal{B}^+, \\ \Lambda(S) - \varepsilon & \text{if } S \in \mathcal{B}^-. \end{cases}$$
   Choose ε to be the maximum value such that Λ remains feasible.
   (a) If some new edge becomes tight, then E^* is extended.
   (b) If $\Lambda(S) = 0$ for some $S \in \mathcal{L} \cap \mathcal{B}^-$ after the modification, then unshrink the node S. Set $\mathcal{L} := \mathcal{L} \setminus S$.

observe that $T \in \mathcal{B}^+$ and further, $\mathcal{B}^+ \cap \mathcal{B}^- = \emptyset$ (otherwise, there exists a $T-T$ alternating walk and hence we should be in case I). The correctness of the output follows immediately due to complementary slackness. We show the termination of the algorithm along very similar lines as the proof of termination of Edmonds' algorithm.

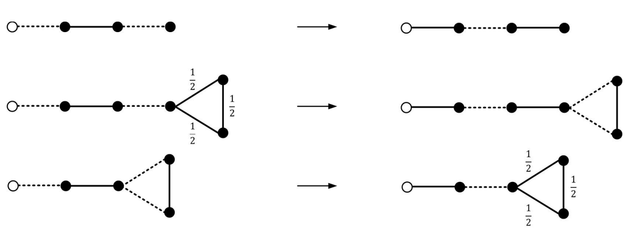

Figure 5. The possible modifications in the Half-integral Matching algorithm.

Let $\beta(z)$ denote the number of exposed nodes plus the number of cycles in supp(z). We first note that $\beta(z) = \beta(z^*)$. This can be derived from Lemma 8(iii). Our next lemma shows that $\beta(z)$ is non-increasing. If $\beta(z)$ is unchanged during a certain number of iterations of the algorithm, we say that these iterations form a *non-decreasing phase*. We say that the algorithm itself is non-decreasing, if $\beta(z)$ does not decrease anytime.

Lemma 15. *Let z be an arbitrary solution during the algorithm, and let α be the number of odd cycles in supp(w) that are absent in supp(z). Then $|W| + odd(w) \geq \beta(z) + 2\alpha$. At termination, $|W| + odd(w) \geq odd(z) + 2\alpha$.*

The non-decreasing scenario: Let us now analyze the first non-decreasing phase $\mathcal{P}$ of the algorithm, starting from the input w. These results will also be valid for later non-decreasing phases as well. Consider an intermediate iteration with z, Λ being the solutions, $\mathcal{L}$ being the laminar family and T being the exposed nodes. Let us define the set of outer/inner nodes of G^* as those having even/odd length alternating walk from R (the set of exposed nodes and node sets of $1/2$-cycles) in G^*. Let $\mathcal{N}_o$ and $\mathcal{N}_i$ denote their sets, respectively. Clearly, $\mathcal{B}^+ \subseteq \mathcal{N}_o$, $\mathcal{B}^- \subseteq \mathcal{N}_i$ in Case II of the algorithm.

Lemma 16. *If $\mathcal{P}$ is a non-decreasing phase, then if a node in $\mathcal{V}^*$ is outer in any iteration of phase $\mathcal{P}$, it remains a node in $\mathcal{V}^*$ and an outer node in every later iteration of $\mathcal{P}$. If a node is inner in any iteration of $\mathcal{P}$, then in any later iteration of $\mathcal{P}$, it is either an inner node, or it has been unshrunk in an intermediate iteration.*

The termination of the algorithm is guaranteed by the following simple corollary.

Corollary 17. *The non-decreasing phase $\mathcal{P}$ may consist of* *at most $|V| + |\mathcal{F}|$ iterations.*

Proof: Case I may occur at most $|W|$ times as it decreases the number of exposed nodes. In Case II, either $\mathcal{N}_i$ is extended, or a set is unshrunk. By Lemma 16, the first scenario may occur at most $|V|$ times and the second at most $|\mathcal{F}|$ times. ■

Lemma 18. *Assume the half-integral matching algorithm is non-decreasing. Let Γ be the initial dual and z, Λ be the terminating solution and $\mathcal{L}$ be the terminating laminar family. Let $\mathcal{N}_o$ and $\mathcal{N}_i$ denote the final sets of outer and inner nodes in G^*.*

- *If $\Lambda(S) > \Gamma(S)$ then S is an outer node in $\mathcal{V}^*$.*
- *If $\Lambda(S) < \Gamma(S)$, then either $S \in \mathcal{F} \setminus \mathcal{L}$, (that is, S was unshrunk during the algorithm and $\Lambda(S) = 0$) or S is an inner node in $\mathcal{V}^*$, or S is a node in $\mathcal{V}^*$ incident to an odd cycle in supp(z).*

B. Proof of convergence

Let us consider two consecutive solutions in Algorithm C-P-Matching. Let x be the unique proper-half-integral optimal solution to $P_{\mathcal{F}}(G,c)$ and Π be an $\mathcal{F}$-positively-critical dual optimal solution to $D_{\mathcal{F}}(G,c)$. We define $\mathcal{H}' = \{S : S \in \mathcal{F}, \Pi(S) > 0\}$ and $\mathcal{H}''$ based on odd cycles in x, and use the critical family $\mathcal{H} = \mathcal{H}' \cup \mathcal{H}''$ for the next iteration. Let y be the unique proper-half-integral optimal solution to $P_{\mathcal{H}}(G,c)$, and let Ψ be an $\mathcal{H}$-positively-critical dual optimal solution to $D_{\mathcal{H}}(G,c)$. We already know that Π is an $\mathcal{H}$-critical dual feasible solution to $D_{\mathcal{H}}(G,c)$ by Lemma 5.

Let us now contract all maximal sets $S \in \mathcal{H}$ with $\Psi(S) > 0$ w.r.t. Ψ to obtain the graph $\hat{G} = (\hat{V}, \hat{E})$ with cost $\hat{c}$. Note that by Lemma 11, Π and Ψ are identical inside S, hence this is the same as contracting w.r.t. Π. Let $\hat{x}$, $\hat{y}$, $\hat{\Pi}$, and $\hat{\Psi}$ be the images of x, y, Π, and Ψ, respectively.

Let $\bar{\mathcal{H}}'' = \{S : S \in \mathcal{H}'', \Psi(S) > 0\}$, and let $W = \cup\bar{\mathcal{H}}''$ denote the union of the members of $\bar{\mathcal{H}}''$. Let $\hat{W}$ denote the image of W. Then $\hat{W}$ is the set of exposed nodes for $\hat{x}$ in $\hat{G}$, whereas the image of every set in $\mathcal{H}'' \setminus \bar{\mathcal{H}}''$ is an odd cycle in $\hat{x}$. Let $\mathcal{N} = \{T \in \mathcal{H}' : T \cap W = \emptyset\}$, $\mathcal{K} = \{T \in \mathcal{N} : \Psi(T) = 0\}$ and $\hat{\mathcal{N}}$ and $\hat{\mathcal{K}}$ be their respective images. All members of $\mathcal{N} \setminus \mathcal{K}$ are contracted to single nodes in $\hat{G}$; observe that $\hat{\mathcal{K}}$ is precisely the set of all sets in $\hat{\mathcal{N}}$ of size at least 3.

We will start the Half-integral Matching algorithm in $\hat{G}$ with $\hat{W}$, from the initial primal and dual solutions $\hat{x}$ and $\hat{\Pi}$. Claim 19(ii) justifies the validity of this input choice for the Half-integral Matching algorithm.

Claim 19. *(i) For every $\hat{\mathcal{L}} \subseteq \hat{\mathcal{K}}$, $\hat{y}$ is the unique optimal solution to $P_{\hat{\mathcal{L}}}(\hat{G},\hat{c})$ and $\hat{\Psi}$ is an optimal solution to $D_{\hat{\mathcal{L}}}(\hat{G},\hat{c})$.*
(ii) $\hat{x}$ is a proper-half-integral optimal solution to $P^{\hat{W}}_{\hat{\mathcal{K}}}(\hat{G},\hat{c})$ and $\hat{\Pi}$ is a $\hat{\mathcal{K}}$-positively-critical dual optimal solution to $D^{\hat{W}}_{\hat{\mathcal{K}}}(\hat{G},\hat{c})$.

Lemma 20. *Suppose we start the Half-integral Primal-Dual algorithm in $\hat{G}$, $\hat{c}$, $\hat{\mathcal{K}}$, $\hat{W}$, from the initial primal and dual solutions $\hat{x}$ and $\hat{\Pi}$. Then the output $\hat{z}$ of the algorithm is equal to $\hat{y}$.*

Proof of Lemma 6: Let us start the Half-integral Matching algorithm in $\hat{G}$, $\hat{c}$, $\hat{\mathcal{K}}$, $\hat{W}$, from the initial primal and dual solutions $\hat{x}$ and $\hat{\Pi}$. Let $\hat{z}$ be the output of the half-integral matching algorithm.

By Lemma 20, $\hat{z} = \hat{y}$. We first observe that odd$(x) = |W| +$ odd$(\hat{x})$. This easily follows by Lemma 8(iii), applied in $G \setminus W$. Let $\alpha = |\mathcal{H}'' \setminus \bar{\mathcal{H}}''|$. There is an odd cycle in supp(x) corresponding to each set of $\mathcal{H}'' \setminus \bar{\mathcal{H}}''$. None of these cycles may be contained in supp$(\hat{z})$ = supp$(\hat{y})$ as otherwise the corresponding cut in $\mathcal{H}''$ would be violated by y. Thus Lemma 15 implies odd$(\hat{y})$ = odd$(\hat{z}) \leq |W| +$ odd$(\hat{x}) - 2\alpha$ and Lemma 8(iii) implies odd(y) = odd$(\hat{y})$. Hence, odd$(y) \leq$ odd$(x) - 2\alpha$. ■

The following claim is a consequence of the above proof.

Claim 21. *If odd(y) = odd(x), then $\mathcal{H}'' = \bar{\mathcal{H}}''$. Further, the Half-integral Matching algorithm applied in $\hat{G}$, $\hat{c}$, $\hat{\mathcal{K}}$, $\hat{W}$, with starting solution $\hat{x}$, $\hat{\Pi}$ is non-decreasing.* □

This claim already implies Lemma 7 for $j = i + 1$. Consider the scenario odd(x) = odd(y). Let us start the half-integral matching algorithm in $\hat{G}$, $\hat{c}$, $\hat{\mathcal{K}}$, $\hat{W}$, from the initial primal and dual solutions $\hat{x}$ and $\hat{\Pi}$. Consider the final dual solution $\hat{\Lambda}$ with corresponding laminar family $\hat{\mathcal{L}}$ and define Λ in G as follows.

If $S \subsetneq T$ for some $T \in \mathcal{H}$, $\Psi(T) > 0$, then set $\Lambda(S) = \Psi(S)$ (this defines the dual solutions for sets and nodes inside T that were contracted to obtain $\hat{G}$). If $\hat{S} \in \hat{\mathcal{L}} \cup \hat{\mathcal{V}}$, then set $\Lambda(S) = \hat{\Lambda}(\hat{S})$ for its pre-image S (this defines the dual solutions for sets and nodes on or outside T that were contracted to obtain $\hat{G}$).

Lemma 22. *Assume odd(x) = odd(y) for the consecutive solutions x and y. Then $\hat{\Lambda} = \hat{\Psi}$ and hence $\Lambda = \Psi$.*

Proof of Lemma 7: Let x_i be the solution in the i'th iteration (above, we used $x = x_i$ and $y = x_{i+1}$). Assume the number of odd cycles does not decrease between iterations i and j. By Claim 21, if we run the half-integral matching algorithm between x_k and x_{k+1}, for $i \leq k < j$, it is always non-decreasing. We first run on the contracted graph $\hat{G} = \hat{G}_i$ starting from primal solution $\hat{x} = \hat{x}_i$ and dual solution $\hat{\Pi} = \hat{\Pi}_i$. Lemmas 20 and 22 show that it terminates with the primal optimal solution $\hat{y} = \hat{x}_{i+1}$ and dual optimal solution $\hat{\Lambda} = \hat{\Psi}$.

For $j = i + 1$, the statement follows by Claim 21 since $\bar{\mathcal{H}}'' = \mathcal{H}''$ means that all cuts added in iteration i have positive dual value in iteration $i + 1$. Further, all sets in $\mathcal{H}''$ were contracted to exposed nodes in $\hat{x}_i$. By Lemma 18, these will be outer nodes on termination of the half-integral matching algorithm as well. Let G^* be the contracted graph upon termination of the Half-Integral Primal-Dual algorithm.

Let $\mathcal{J} = \mathcal{J}' \cup \mathcal{J}''$ be the set of cuts imposed in the $(i+2)$'th round, with $\mathcal{J} = \{S \in \mathcal{H} : \Psi(Z) > 0\}$, and let $\mathcal{J}''$ be defined according to odd cycles in x_{i+1}. Let Φ be the extremal dual optimal solution to $D_{\mathcal{J}}(G, c)$.

Let us run the half-integral matching algorithm from x_{i+1} to x_{i+2}. We start the algorithm with the contracted graph $\hat{G}_{i+1}$, which results by contracting all sets with $\Phi(S) > 0$, $S \in \mathcal{J}$. Let $\hat{G}^*_{i+1}$ be the initial contraction of $\hat{G}_{i+1}$ used by the algorithm.

The key observation is that while the underlying graphs $\hat{G}_i$ and $\hat{G}_{i+1}$ are different, $\hat{G}^*_{i+1}$ can be obtained from G^* by contracting those odd cycles corresponding to the sets of $\mathcal{J}''$. Every other node that was inner or outer node in G^* will also be inner or outer node in $\hat{G}^*_{i+1}$, including the members of $\mathcal{H}''$. By Lemma 18, the members of $\mathcal{H}''$ will be outer nodes at termination, along with the new outer nodes $\mathcal{J}''$.

Iterating this argument one can show that every set that was imposed based on an odd cycle between iterations i and k will be outer nodes at the termination of the Half-integral Matching algorithm from x_k to x_{k+1}. ∎

VII. Open Questions

Our initial motivation was to bound the number of iterations of the cutting plane method using the Padberg-Rao procedure. This question remains open and any analysis would have to deal with non-half-integral solutions.

Given the encouraging results of this paper, it would be interesting to prove efficient convergence of the cutting plane method for other combinatorial polytopes. For example, one could try a similar approach for finding an optimal solution for b-matchings. Another direction could be to try this approach for optimizing over the subtour elimination polytope.

Acknowledgment

We are grateful to William Cook and Richard Karp for their kind help and encouragement.

References

[1] M. Padberg and M. Rao, "Odd minimum cut-sets and b-matchings," *Mathematics of Operations Research*, vol. 7, no. 1, pp. 67–80, 1982.

[2] M. Grötschel and O. Holland, "Solving matching problems with linear programming," *Mathematical Programming*, vol. 33, no. 3, pp. 243–259, 1985.

[3] L. Lovász and M. Plummer, *Matching theory*. North Holland, 1986.

[4] M. Trick, "Networks with additional structured constraints," Ph.D. dissertation, Georgia Institute of Technology, 1987.

[5] M. Fischetti and A. Lodi, "Optimizing over the first Chvátal closure," *Mathematical Programming*, vol. 110, no. 1, pp. 3–20, 2007.

[6] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization*. Wiley-Interscience, 1999.

[7] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & sons, 1998.

[8] R. Kannan, "Minkowski's convex body theorem and integer programming," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 415–440, 1987.

[9] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Operations Research*, vol. 2, no. 4, pp. 393–410, Nov 1954.

[10] R. Gomory, "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathematical Society*, vol. 64, no. 5, pp. 275–278, 1958.

[11] ——, "Solving linear programs in integers," in *Proceedings of Symposia in Applied Mathematics*, vol. 10, July 1960, pp. 211–215.

[12] ——, "An algorithm for integer solutions to linear programs," *Recent advances in mathematical programming*, p. 269, 1963.

[13] E. Balas, "Intersection Cuts - A New Type of Cutting Planes for Integer Programming," *Operations Research*, vol. 19, no. 1, pp. 19–39, Jan 1971.

[14] V. Chvátal, "Edmonds polytopes and a hierarchy of combinatorial problems," *Discrete Mathematics*, vol. 4, no. 4, pp. 305–337, Apr 1973.

[15] E. Balas, *Disjunctive Programming*. Elsevier, 1979, vol. 5, pp. 3–51.

[16] W. Cook, R. Kannan, and A. Schrijver, "Chvátal closures for mixed integer programming problems," *Mathematical Programming*, vol. 47, no. 1, pp. 155–174, May 1990.

[17] G. L. Nemhauser and L. A. Wolsey, "A recursive procedure to generate all cuts for 0-1 mixed integer programs," *Mathematical Programming*, vol. 46, no. 1, pp. 379–390, Jan 1990.

[18] E. Balas, S. Ceria, and G. Cornuéjols, "A lift-and-project cutting plane algorithm for mixed 0-1 programs," *Mathematical Programming*, vol. 58, no. 1, pp. 295–324, Jan 1993.

[19] L. Lovász and A. Schrijver, "Cones of matrices and set-functions and 0-1 optimization," *SIAM Journal of Optimization*, vol. 1, pp. 166–190, 1991.

[20] H. D. Sherali and W. P. Adams, "A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems," *Discrete Applied Mathematics*, vol. 52, no. 1, pp. 83–106, Jul 1994.

[21] G. Cornuéjols and Y. Li, "Elementary closures for integer programs," *Operations Research Letters*, vol. 28, no. 1, pp. 1–8, Feb 2001.

[22] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[23] J. Edmonds, "Maximum matching and a polyhedron with 0, 1-vertices," *Journal of Research of the National Bureau of Standards*, vol. 69, no. 1-2, pp. 125–130, 1965.

[24] A. Schrijver, *Combinatorial optimization*. Springer, 2003.