Mathematical Programming 65 (1994) 139–150

A separation algorithm for the matchable set polytope

William H. Cunningham*[,a], Jan Green-Krótki[b]

[a]*Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1*
[b]*19, rue de l'Ecole Polytechnique, 75005 Paris, France*

Received 19 June 1989; revised manuscript received 12 August 1993

Abstract

A matchable set of a graph is a set of vertices joined in pairs by disjoint edges. Balas and Pulleyblank gave a linear-inequality description of the convex hull of matchable sets. We give a polynomial-time combinatorial algorithm for the separation problem for this polytope, and a min–max theorem characterizing the maximum violation by a given point of an inequality of the system.

Keywords: Combinatorial optimization; Polyhedra; Matching; Matchable set; Separation algorithm; Augmenting path

1. Introduction

A matchable set of a graph $G=(V,E)$ is a set of vertices joined in pairs by disjoint edges. Balas and Pulleyblank gave a characterization by linear inequalities of the convex hull Q of incidence vectors of matchable sets of G. The main result of this paper is an efficient algorithm that, given $x\in\mathbb{R}^V$, either determines that $x\in Q$ or finds a linear inequality satisfied by every point of Q and violated by x.

Our algorithm actually finds, in the second case, an inequality in the Balas–Pulleyblank system that is most-violated by x. The algorithm proves a min–max theorem characterizing the amount of this violation. It states that the maximum violation is the minimum "deficiency" of points $y\in Q$ such that $y\leqslant x$. Such a minimizing y can be required to satisfy additional discreteness conditions. The min–max theorem generalizes the Balas–Pulleyblank result as well as the Tutte–Berge matching formula.

*Corresponding author. Part of this work was done while this author was visiting the Institut für Operations Research and the Forschungsinstitut für Diskrete Mathematik, Universität Bonn, Germany. Research supported by SFB303 DFG, Germany and by NSERC of Canada.

SSDI 0025-5610(93)E0079-T

In this paper we use standard graph terminology. For $x \in \mathbb{R}^S$ and $T \subseteq S$, $x(T)$ denotes $\sum_{j \in T} x_j$. Where $G=(V, E)$ is a graph and A, B are subsets of V, we use $[A, B]$ to denote $\{e \in E$: e joins an element of A to an element of $B\}$. In particular, $\delta(A)$ denotes $[A, V \backslash A]$ and $\gamma(A)$ denotes $[A, A]$. We often omit brackets; for example, for $v \in V$, we may write $\delta(v)$ instead of $\delta(\{v\})$ and for $A, B \subseteq V$, $x(A, B)$ instead of $x([A, B])$. Where graphs other than G are in use, it may be necessary to subscript graph notation, for example, $\delta_G(A)$. Where a ground set S is understood (usually V or E), the incidence vector of a subset T of S is denoted χ^T.

A *matching* in $G=(V, E)$ is a set of edges of G, no two having a common end. Edmonds (1965) gave a characterization of $P=P(G)=\text{conv}(\{\chi^R$: R a matching of $G\}) \subseteq \mathbb{R}^E$ by linear inequalities. We call P the *matching polytope* of G. He also gave a polynomial-time algorithm to find an optimal weight matching, or equivalently, to optimize any linear function over P.

Theorem 1.1 (Edmonds). *The matching polytope of a graph $G=(V, E)$ is the set of all $z \in \mathbb{R}^E_+$ satisfying*
$z(\delta(v)) \leqslant 1$, *for each* $v \in V$;
$z(\gamma(S)) \leqslant \frac{1}{2}(|S|-1)$ *for each* $S \subseteq V$ *such that* $|S|$ *is odd and at least* 3.

A matchable set in G is a set of vertices constituting the ends of the edges of some matching. (Hence, in particular, every matchable set has even cardinality.) Balas and Pulleyblank (1989) gave a characterization of $Q=Q(G)=\text{conv}\{\chi^T$: T a matchable set$\} \subseteq \mathbb{R}^V$ by linear inequalities. We call Q the *matchable set polytope* of G.

Theorem 1.2 (Balas-Pulleyblank). *The matchable set polytope of a graph $G=(V, E)$ is the set of all $x \in \mathbb{R}^V_+$ satisfying*
$x_v \leqslant 1$, *for each* $v \in V$;
$x(\bigcup_1^k A_i) - x(B) \leqslant \sum_{i=1}^k (|A_i|-1)$, *for all subsets* $B, A_1, \ldots, A_k$ *of* V,
such that each A_i is the vertex-set of an odd component of $G-B$.

Notice that T is a matchable set if and only if $\chi^T = M\chi^R$ for some matching R, where $M \in \mathbb{R}^{V \times E}$ is the incidence matrix of G. This relationship carries over to convex combinations, so $Q=\{Mz$: $z \in P\}$, that is, Q is a linear transformation of P. We call the vector Mz, for $z \in \mathbb{R}^E$, the *degree sequence* of z. (The terminology is suggested by the special case in which z is the incidence vector of the edge-set of a subgraph of G, in particular, where G is a complete graph.)

We remark that Theorem 1.1 has been generalized to characterize the convex hull of "capacitated b-matchings" by Edmonds and Johnson (1970), and Theorem 1.2 has been similarly generalized to the convex hull of their degree sequences; see Cunningham and Green-Krótki (1991).

Notice that the above linear relationship between P and Q implies that there is an easy way to optimize any linear function over Q. Namely, $\max(cx$: $x \in Q) = \max((cM)z$: $z \in P)$, so optimization over Q reduces to a weighted matching problem. In addition to this *optimization* problem, a second fundamental problem for the polyhedron Q is the *separation* problem: Given $x \in \mathbb{R}$, find, if one exists, a linear inequality satisfied by every point in Q but violated by x. Due to a certain equivalence between the two problems, the separation problem for Q is solvable via the ellipsoid method; see Grötschel et al. (1981). However, a more direct combinatorial method is desirable. The separation algorithm in this paper is combinatorial, runs in polynomial time, and gives an independent proof of Theorem 1.2. (For P, such a separation algorithm has been found by Padberg and Rao (1982), but their algorithm does not provide a proof of Theorem 1.1.)

We shall solve the separation problem for Q in the following form. Given $x \in \mathbb{R}^V_+$, $0 \leqslant x \leqslant 1$, find, if there is one, a constraint Theorem 1.2(b) that is most-violated by x. In fact, there is a min–max theorem characterizing this maximum violation. (Notice that $\min(\sum_1^k(|A_i|-1) - (x\bigcup_1^k A_i) - x(B)))$ differs from the minimum in Theorem 1.3 below by the constant $x(V)$.)

Theorem 1.3. *Let* $x \in \mathbb{R}^V$, $0 \leqslant x \leqslant 1$. *Then*

$$\max(y(V): y \leqslant x,\ y \in Q) = \min\left(\sum_1^k (|\mathrm{A}_i|-1) + x(B) + x(V \setminus \bigcup_1^k A_i)\right),$$

where the minimum is taken over all subsets B of V and vertex-sets $A_1, \ldots, A_k$ of odd components of $G-B$.

If we call $x(V) - y(V)$ the *deficiency* of y (with respect to x), then Theorem 1.3 asserts "max violation = min deficiency". Actually, this result also extends the Balas–Pulleyblank Theorem 1.2. Namely, $x \in Q$ if and only if the maximum is $x(V)$, which is true by Theorem 1.3 if and only if

$$x(V) \leqslant \sum_1^k (|A_i|-1) + x(B) + x\left(V \setminus \bigcup_1^k A_i\right)$$

for all choices of B, $A_1, \ldots, A_k$, thus proving Theorem 1.2. A further strengthening of Theorem 1.3, on the discreteness of the maximizing y, is useful in establishing the efficiency of our algorithm.

Theorem 1.4. *There is a maximizing y in Theorem* 1.3 *such that each y_v is an integral combination of elements of* $\{x_v: v \in V\} \cup \{1\}$.

Applying Theorem 1.4 to the case in which each x_v is an integer multiple of some number α, we conclude that there is also a maximizing y having this property. In particular, if each x_v is 0 or 1, then there is a $\{0, 1\}$-valued maximizing y. So the left-hand side of the equation in Theorem 1.3 is the maximum size of a matchable set, and the right-hand side is

$\min(|V|-k)$; that is, the Tutte–Berge matching formula is a consequence of Theorems 1.3 and 1.4.

We can also say something about the existence of a discrete $z \in P$ such that a given $y \in Q$ is its degree sequence.

Theorem 1.5. *If $y \in Q$, then there exists $z \in P$ such that y is the degree sequence of z and for each $e \in E$, $2z_e$ is an integer combination of $\{y_v: v \in V\} \cup \{1\}$.*

To see that the factor of 2 in Theorem 1.5 is needed, consider the case in which G is a triangle and each y_v is $\frac{1}{2}$. However, for G bipartite the factor of 2 is not needed, and in fact this version of Theorem 1.5 follows easily from well-known results. In the bipartite case, $y \in Q$ if and only if $0 \leqslant y \leqslant 1$ and there exists $z \in \mathbb{R}_+^E$ such that $Mz=y$, where M is the incidence matrix of G. Since in this case M is totally unimodular, z can be chosen so that each z_e is an integral combination of $\{y_v: v \in V\} \cup \{1\}$. From similar considerations we can also easily obtain Theorem 1.4 for bipartite G. (Notice that *both* the characterization of membership in Q *and* the total unimodularity of M fail for G non-bipartite.)

The general approach of the separation algorithm is as follows. We keep $z \in P$ and $y = Mz \in Q$ with $y \leqslant x$, and try to increase $y(V) = 2z(E)$ by augmenting-path techniques. The techniques used are mostly familiar ones from matching theory. However, whereas in usual matching algorithms an augmentation obviously maintains feasibility, here we must explicitly watch for violations of $z(\gamma(S)) \leqslant \frac{1}{2}(|S|-1)$. Similarly, whereas in usual unweighted matching algorithms the objective function increases by an integer with each augmentation, here the increase can be small. We shall handle the first difficulty by requiring z to have properties that strongly limit the search for violated inequalities. For the second difficulty, discreteness properties like Theorems 1.4 and 1.5 will be useful.

2. Sufficient conditions for membership in P

Our algorithm maintains $z \in P$, but it is by no means trivial to recognize membership in the matching polytope. However, we are able to certify membership in P for vectors z that satisfy certain additional conditions, which can be required in the separation algorithm. In this section we establish these sufficient conditions for membership in P.

We introduce here further terminology. Much of it was first presented in Pulleyblank (1973). A family of $\mathscr{S}$ of subsets of V is *nested* if there do not exist $\mathscr{S}_1, \mathscr{S}_2 \in \mathscr{S}$ with $\mathscr{S}_1 \backslash \mathscr{S}_2$, $\mathscr{S}_2 \backslash \mathscr{S}_1$, $S_1 \cap S_2$ all non-empty. Given a nested family $\mathscr{S}$, the graph $G \times \mathscr{S}$ is obtained by shrinking each maximal member S of $\mathscr{S}$ to a single vertex, which we call S, and deleting all edges in $\gamma(S)$. Since the maximal sets are disjoint, this graph is well defined. Notice that every vertex of $G \times \mathscr{S}$ is an element or a subset of V, and that every element of V is either a vertex of $G \times \mathscr{S}$ or an element of a (unique) vertex of $G \times \mathscr{S}$. The first kind of vertex is called *real*, the second *pseudo*. For a nested family $\mathscr{S}$ and $S \in \mathscr{S}$, we denote by $\mathscr{S}[S]$ the family $\{S'\colon S \supset S' \in \mathscr{S}\}$. We use $G[S]$ to denote the subgraph of G induced by S. A *shrinking* family of G is a nested family $\mathscr{S}$ such that:

For all $S \in \mathscr{S}$, $G[S] \times \mathscr{S}[S]$ is spanned by an odd cycle $C(S)$.

It follows that if $\mathscr{S}$ is a shrinking family, then every pseudo vertex of $G \times \mathscr{S}$ has odd cardinality.

The next two results give ways to ensure that a given (fractional) z is in P. For $z \in \mathbb{R}^E_+$, we use $G_+ = G_+(z)$ to denote the spanning subgraph of G whose edge-set is $\{j \in E\colon z_j > 0\}$. We say that a graph is *independent* if it has no even cycle and each component contains at most one odd cycle. (The term arises from considering linear independence of column submatrices of the incidence matrix of G.)

Lemma 2.1. *If* $z \in \mathbb{R}^E_+$, $z(\delta(v)) \leqslant 1$ *for all* $v \in V$, G_+ *is independent, and* $z(\gamma(S)) \leqslant \frac{1}{2}(|S|-1)$ *whenever* S *is the vertex-set of an odd cycle, then* $z \in P(G)$.

Proof. If not, let S be a minimal set violating $z(\gamma(S)) \leqslant \frac{1}{2}(|S|-1)$. It it easy to see that $G_+[S]$ is connected. If there exists $v \in S$ such that v is adjacent in $G_+[S]$ to a degree-one vertex w, then

$$z(\gamma(S\setminus\{v, w\})) \geqslant z(\gamma(S)) - z(\delta(v)) \geqslant z(\gamma(S)) - 1 > \tfrac{1}{2}(|S\setminus\{v, w\}| - 1)$$

a contradiction. Therefore, since $G_+[S]$ is independent, it must consist of an odd cycle, also a contradiction. □

Lemma 2.2. *Suppose that* $S \subseteq V$ *is spanned by an odd cycle* $C(S)$ *in* G *and* $z' \in P(G')$ *where* $G' = G \times \{S\} = (V', E')$. *Then* z' *can be extended to* $z \in \mathbb{R}^E$ *such that* $z \in P(G)$ *and* $z(\gamma(S)) = \frac{1}{2}(|S|-1)$. *Moreover, if* $x \in \mathbb{R}^V$ *and* x, z' *satisfy* $0 \leqslant x \leqslant 1$, $z'(\delta_{G'}(v)) \leqslant x_v$, $v \in V'\setminus\{S\}$, *and* $z'(\delta(S)) \leqslant x(S) - (|S|-1)$, *then* z *can be chosen also to satisfy* $z(\delta(v)) \leqslant x_v$, $v \in V$.

Proof. We prove the second statement only, since the first is equivalent to the second when each $x_v = 1$. Pick a vertex $w \in S$, define b_w to be $x_w - z'(\delta(w) \cap E') + z'(\delta(S)) + |S| - 1 - x(S)$, and define b_v, for $v \in S\setminus\{w\}$, to be $x_v - z'(\delta(v) \cap E')$. The system $z(\delta_{G[S]}(v)) = b_v$ for each $v \in S$, and $z_j = 0$ for each $j \in \gamma(S)\setminus E(C(S))$, has a unique solution for z_j, $j \in \gamma(S)$. The resulting $z \in \mathbb{R}^E$ certainly satisfies $z(\delta(v)) \leqslant x_v$ for $v \neq w$. Moreover,

$$x_w - z(\delta(w)) = x_w - z'(\delta(w) \cap E') - b_w = x(S) - z'(\delta(S)) - (|S|-1) \geqslant 0.$$

To show that z is also non-negative, we use the following formula for z_j, $j \in E(C(S))$. Suppose that the vertex-sequence of $C(S)$ is $v_0, v_1, \ldots, v_{|S|-1}, v_0$, and let j be the edge from v_0 to $v_{|S|-1}$. Then $z_j = \frac{1}{2}(\Sigma(b_{v_i}\colon i \text{ even}) - \Sigma(b_{v_i}\colon i \text{ odd}))$. Therefore, since $b(S) = |S| - 1$, $2z_j = |S| - 1 - 2\Sigma(b_{v_i}\colon i \text{ odd})$. Now if w has an even subscript, then $2z_j \geqslant |S| - 1 - 2\Sigma(x_{v_i}\colon i \text{ odd})$. If w has an odd subscript, then

$$2z_j = |S| - 1 - 2\Sigma(x_{v_i}\colon i \text{ odd}) + 2\Sigma(z'(\delta(v_i) \cap E')\colon i \text{ odd})$$
$$+ 2z'(\delta(S)) - 2(|S| - 1) + 2x(S)$$
$$\geqslant |S| - 1 - 2\Sigma(x_{v_i}\colon i \text{ odd}).$$

In both cases, since there are $\frac{1}{2}(|S| - 1)$ odd subscripts and each $x_v \leqslant 1$, we have $z_j \geqslant 0$.

Finally, we need to show that $z(\gamma(S')) \leqslant \frac{1}{2}(|S'| - 1)$ for all odd subsets S' of V. If $S' \subseteq S$, this is true by Lemma 2.1, since $z(\gamma(S)) = \frac{1}{2}(|S| - 1)$ and if $S' \supseteq S$ it follows from

$$\begin{aligned} z(\gamma(S')) &= z'(\gamma_{G'}((S' \backslash S) \cup \{S\})) + z(\gamma(S)) \\ &\leqslant \tfrac{1}{2}|S' \backslash S| + \tfrac{1}{2}(|S| - 1) = \tfrac{1}{2}(|S'| - 1). \end{aligned}$$

So suppose that $S' \backslash S$, $S \backslash S'$, $S \cap S'$ are all non-empty. If $|S \cap S'|$ is odd, then

$$\begin{aligned} z(\gamma(S')) + z(\gamma(S)) &\leqslant z(\gamma(S \cup S')) + z(\gamma(S \cap S')) \\ &\leqslant \tfrac{1}{2}(|S \cup S'| - 1) + \tfrac{1}{2}(|S \cap S'| - 1) \\ &= \tfrac{1}{2}(|S'| - 1) + \tfrac{1}{2}(|S| - 1) \\ &= \tfrac{1}{2}(|S'| - 1) + z(\gamma(S)), \end{aligned}$$

so $z(\gamma(S')) \leqslant \frac{1}{2}(|S'| - 1$. If $|S \cap S'|$ is even, then

$$\begin{aligned} z(\gamma(S')) + \tfrac{1}{2}(|S| - 1) &= z(\gamma(S')) + z(\gamma(S)) \\ &\leqslant z(\gamma(S \backslash S')) + z(\gamma(S' \backslash S)) + \Sigma(z(\delta(v))\colon v \in S \cap S') \\ &\leqslant \tfrac{1}{2}(|S \backslash S'| - 1) + \tfrac{1}{2}(|S' \backslash S| - 1) + |S \cap S') \\ &= \tfrac{1}{2}(|S| - 1) + \tfrac{1}{2}(|S'| - 1) \end{aligned}$$

so $z(\gamma(S')) \leqslant \frac{1}{2}(|S'| - 1)$, as required. □

For a shrinking family $\mathscr{S}$ of G, a vector $x \in \mathbb{R}^V$ induces a vector $x' \in \mathbb{R}^{V'}$, where $G \times \mathscr{S} = G' = (V', E')$. Namely, $x'_v = x_v$ for v a real vertex of G' and $x'_S = x(S) - (|S| - 1)$ for S a pseudo vertex of G'. By a slight abuse of notation, we also use x to denote this x'. We call $z \in P(G)$ *feasible* if $z(\delta(v)) \leqslant x_v$ for each $v \in V$. The *deficiency* of $v \in V$ (with respect to a feasible z), is $x_v - z(\delta(v))$; v is *deficient* if it has positive deficiency. The *deficiency* of z is $\Sigma(x_v - z(\delta(v))\colon v \in V)$; that is, it is the deficiency of the degree sequence y of z, as defined earlier. Now Lemma 2.2 implies that, if $z \in P(G \times \mathscr{S})$ is feasible, then it can be extended to a feasible $z \in P(G)$ having the same deficiency.

One more remark will be useful later. Although the process of extending z requires some work, extending its degree sequence y is easy. Namely, for each pseudo vertex S of G' we choose some $w \in S$ and define y_w to be $x_w + y_S - x(S) + |S| - 1$, and y_v to be x_v for each $v \in S \backslash \{w\}$.

3. Augmenting paths

In this section we introduce the basic methods for finding an improvement in the current y. In fact, we work mainly with z in the matching polytope.

In a path $v_0, e_1, v_1, e_2, \ldots, e_k, v_k$ a vertex or edge is *even* or *odd* according to the parity of its subscript in its first occurrence as a vertex or edge term in the sequence. In a tree T with distinguished *root* vertex r, a vertex (or edge) is *even* or *odd* according to its parity in a simple path in T beginning at r. Similarly, we define *even* and *odd* for vertices of a forest of rooted trees. Given $z \in P(G')$, a path in G' is *z-alternating* if $z_j > 0$ for every even edge of the path. If z is also feasible with respect to $x \in \mathbb{R}_+^V$, a path is *z-augmenting* if it is z-alternating and its end-vertices are deficient. We deal with three types of augmenting path. A *type 1* augmenting path is one that is simple. (This is the only type of augmenting path that arises in the special case of maximum cardinality matching.) One of *type 2* is of the form $v_0, e_1, v_1, \ldots, e_{4k+2l+1}$ for some $k \geqslant 0$, $l \geqslant 1$ where $v_0, \ldots, v_{2k+2l}$ are distinct vertices, $v_{4k+2l+1-i} = v_i$ for $0 \leqslant i \leqslant 2k$, and $e_{4k+2l+2-i} = e_i$ for $1 \leqslant i \leqslant 2k$; that is, it consists of an even-length path, traversed once in each direction, together with an odd-length cycle. An augmenting path of *type 3* is of the form $v_0, e_1, \ldots, e_{4k+2l-1}, v_{4k+2l-1}$ for some $k \geqslant 1$, $l \geqslant 1$, where $v_0, \ldots, v_{2k+2l-1}$ are distinct vertices, $v_{4k+2l-1-i} = v_i$ for $0 \leqslant i \leqslant 2k-1$, and $e_{4k+2l-i} = e_i$ for $1 \leqslant i \leqslant 2k-1$; that is, it consists of an odd-length path, traversed once in each direction, together with an odd-length cycle. *Augmenting* z by *amount* $\varepsilon \geqslant 0$ on a z-augmenting path P, means increasing z_j by ε for each odd edge occurring once in P and by 2ε for each odd edge occurring twice, and similarly lowering z_j for even edges. It is easy to see that an augmentation changes y only at the ends of the augmenting path, and that $z(E')$ increases by ε.

The amount ε is calculated as follows. For a type 1 path, $\varepsilon \leqslant z_j$ for each even edge j, and ε is at most the deficiency of the end vertices. For a type 2 path, $2\varepsilon \leqslant z_{e_{2i}}$ for $1 \leqslant i \leqslant k$, $\varepsilon \leqslant z_{e_{2i}}$ for $k+1 \leqslant i \leqslant k+l$, 2ε is at most the deficiency of v_0, and $\varepsilon \leqslant \frac{1}{2}(|V(C)| - 1) - z(E(C))$, where C is the odd cycle contained in P. For a type 3 path, $2\varepsilon \leqslant z_{e_{2i}}$ for $1 \leqslant i \leqslant k-1$, $\varepsilon \leqslant z_{e_{2i}}$ for $k \leqslant i \leqslant k+l$, and 2ε is at most the deficiency of v_0. In all cases ε is the largest number satisfying the upper bounds. By using the augmentations only in situations in which Lemma 2.1 can be applied, we can be sure that they maintain $z \in P(G')$.

The algorithm requires the following properties of $G'_+ = G'_+(z)$:

(3.1a) G'_+ is independent;
(3.1b) each component of G'_+ has at most one deficient vertex;
(3.1c) if a component of G'_+ has a deficient vertex, then it has no cycle.

It searches for augmentations by growing in G' a forest F satisfying (3.2) below. A rooted tree T is *z-alternating* if every simple path in T beginning at its root is z-alternating.

(3.2a) No edge of G'_+ has exactly one end in F;
(3.2b) the roots of trees in F are precisely the deficient vertices of G';
(3.2c) each tree in F is z-alternating.

In order to motivate the steps of the separation algorithm, let us first describe its terminating conditions. Suppose that

(3.3a) No odd vertex of F is pseudo;

(3.3b) every edge incident in G' with an even vertex of F is incident also with an odd vertex of F.

We let B be the set of odd vertices of F and let each even pseudo vertex be an A_i and each real even vertex be a singleton A_i. Then the A_i are vertex-sets of odd components of $G-B$. Extend the current z to $\mathbb{R}^E$ and let y be its degree sequence. Then $z(\gamma(A_i)) = \frac{1}{2}(|A_i|-1)$ for all i. Moreover, $v \in A_i$, j incident to v in G, and $z_j > 0$ imply that $j \in \gamma(A_i)$ or $j \in [A_i, B]$. Similarly, j incident to $v \in B$ and $z_j > 0$ imply that $j \in [\bigcup_i A_i, B]$. Hence $y(B) = z(B, \bigcup_i A_i)$, and $y(A_i) = |A_i| - 1 + z(A_i, B)$. Since all deficient vertices are in some A_i, the deficiency of y is $x(\bigcup_i A_i) - y(\bigcup_i A_i) = x(\bigcup_i A_i) - x(B) - \Sigma_i(|A_i|-1)$, which is the amount by which x violates the corresponding inequality in Theorem 1.2. (Notice that the above argument also handles the special case in which $x = y$; then B, as well as each A_i, is empty.) So (3.3) is a valid termination condition.

We now describe an algorithm that, given $\mathscr{S}$, $G' = G \times \mathscr{S}$, and feasible $z \in P(G')$ with at least one deficient vertex, finds either an augmentation of positive amount or encounters the termination conditions (3.3). The algorithm requires of $\mathscr{S}$, z the conditions (3.1) throughout. In addition it requires initially the condition:

(3.4) For each non-deficient pseudo vertex S of G', the extension of z obtained when S is expanded satisfies $z_j > 0$ for each $j \in E(C(S))$.

Given $\mathscr{S}$, z satisfying (3.1) but violating (3.4), one expands the offending S, and clearly this does not introduce violations of (3.1). Hence this additional requirement can be met. We say that an odd cycle C is *tight* (with respect to z) if $z(E(C)) = \frac{1}{2}(|V(C)|-1)$.

Algorithm Augment. Successively expand pseudo vertices S violating (3.4) until (3.4) is satisfied. Let F be the forest consisting of the components of G'_+ containing the deficient vertices. Now if the termination conditions (3.3) are not met, we have the following cases.

(3.5) (*Type* 1 *augmentation.*) There is an edge e joining in G' even vertices u, v of different trees T_1, T_2 of F. Let r_1, r_2 be the roots of T_1, T_2. Then the path in T_1 from r_1 to u, together with e, together with the path in T from v to r_2, forms an augmenting path of type 1. Perform the augmentation and terminate.

(3.6) (*Type 2 or 3 augmentation and shrinking.*) There is an edge e joining in G' even vertices u, v of the same tree T of F. Let r be the root of T. Then the path in T from r to u, together with e, together with the path in T from v to r, forms an augmenting path of type 2 or 3. Perform the augmentation. If the corresponding odd cycle C is now tight, shrink C into a new pseudo vertex S. If the augmentation was of non-zero amount, terminate. If not, update F by deleting from T the edges no longer in G'.

(3.7) (*Simple forest extension.*) There is an edge e joining in G' an even vertex of a tree T of F to a vertex v, such that the component of G'_+ containing v is a tree T_1 not in F. Add e and T_1 to T.

(3.8) (*Non-simple forest extension, augmentation, shrinking.*) There is an edge e joining in G' an even vertex u of a tree T to a vertex v, such that the component H of G'_+ containing v is not a tree. Add e and H to T and delete from T an edge j of the cycle C in H, so that in the new T, j joins two even vertices. Now apply (3.6) with j replacing e.

(3.9) (*Expanding an odd vertex, augmentation.*) There is an odd vertex S of a tree T of F that is a pseudo vertex. Let r be the root of T. Expand S. Then the path in T from r to S, together with $C(S)$, together with the path in T from S to r, forms an augmenting path of type 3. Perform the augmentation and terminate.

Theorem 3.10. *Algorithm Augment terminates in time* $O(|V|^2)$, *either with* $B, A_1, \ldots, A_k$ *satisfying the equality in Theorem* 1.3 *with the degree sequence* y *of* z, *or with a new* $z \in P(G)$ *having larger component-sum and satisfying* (3.1).

Proof. First, we check that, in the steps that do not lead to termination, the properties (3.2) required of F are maintained. This is easy to see for a simple forest extension step. Now consider a non-simple forest extension step (3.8). This leads to an augmentation of amount zero only if the augmentation is of type 2, and the cycle C is tight, and hence is shrunk. Then since all edges added to T except e have $z_e > 0$ and e is odd in the new tree, the tree remains z-alternating. Finally, consider the case of step (3.6). Again if the augmentation is of amount 0, then it is of type 2 and the cycle C is tight and hence is shrunk. However, in this case we have the additional complication that odd vertices v of T have been shrunk into a new even vertex S. Then any vertex w of T whose path to r used an edge j incident to v but not in $E(C)$, would change its parity in T after the shrinking. However, we show that no such edge j exists. Since the augmentation has zero amount, z_e remains zero and so we have $\frac{1}{2}(|V(C)|-1) = z(E(C)) \leqslant \Sigma(x_v\colon v \in V(C),\ v \text{ odd in } T) \leqslant \frac{1}{2}(|V(C)|-1)$. It follows that each such v has $x_v = 1$ and has degree 2 in T, as required.

We now know that the algorithm keeps F satisfying (3.2) and that even vertices remain even, or are shrunk inside larger even vertices. The latter fact implies that any odd pseudo vertex of F must have been a pseudo vertex when G'_+ satisfied (3.4), although even pseudo vertices can violate this condition. Therefore, it is true that the expansion step (3.9) does lead to an augmentation of non-zero amount. To show that the algorithm terminates in polynomial time, consider the number $p(F) = |\{v\colon v$ is a real even vertex of F, or is an element of an even pseudo vertex of F, or is an odd vertex of F with $x_v = 0\}|$. It is easy to check that each of the three types of steps that do not lead to termination increases $p(F)$ by at least 1. Hence there can be at most $|V|$ such steps before termination occurs. The claimed running time is now straightforward to verify, using familiar ideas from matching algorithms.

Finally, we need to check that the properties (3.1) required of G'_+ are maintained. This is easy to see, since the definition of the amount of an augmentation ensures that some z_j

becomes 0 or some vertex becomes non-deficient or some odd cycle is shrunk, correcting any temporary violation of (3.1). □

4. The separation algorithm

We describe two separation algorithms for Q. The first is not efficient, but it is simple, and it can be used to prove the theorems stated in Section 1. The second is not so simple because it uses scaling, but it runs in polynomial time. Of course, both algorithms depend on Augment.

The primitive separation algorithm is just this: Begin with $\mathscr{S} = \phi$, $z = 0$ and successively apply Augment, terminating the main algorithm only if Augment terminates with conditions (3.3) satisfied. (Then we can extend y, and z if desired, from G' to all of G.) We claim that only a finite number of vectors z can occur during the algorithm. The reason is that z is uniquely determined from knowledge of S, G'_+ and the deficient vertices of G'. (This fact is well-known, but also easy to verify.) Since the deficiency of z is decreased by each augmentation of non-zero amount, it follows that Augment is used only a finite number of times, and so the separation algorithm is finitely terminating.

The max violation-min deficiency Theorem 1.3 is an immediate consequence of the finiteness of the algorithm. The discreteness Theorems 1.4 and 1.5 are proved by observing that the algorithm maintains y and z with the required properties. That augmentations preserve these properties is easy to check. That expansion does also is obvious in the case of y, and for z is a consequence of the formula for extending z in the proof of Lemma 2.2.

As usual, the scaling algorithm will work by solving separation problems for a sequence of values of x, at each step using the answer to the last problem to begin the solution of the next. The way in which successive problems change is simple; it will be enough for us to suppose that x' has been obtained from x by increasing one x_v by a number α. Suppose that we have solved the problem for G and x, and have $\mathscr{S}$, z, y, B, A_1, …, A_k. If $v \in \bigcup A_i$ then the deficiency of y with respect to x' is exactly the violation of a constraint in Theorem 1.2 by x', and so $\mathscr{S}$, z, y, B, A_1, …, A_k solve the new problem as well. Otherwise, we want to apply the separation algorithm to solve the new problem beginning with $\mathscr{S}$, z. However, v or the pseudo vertex containing v has become deficient, so the properties (3.1) required of G'_+ may be violated. There are two possibilities for such a violation:

(4.1) There are now two deficient vertices in the same (tree) component T of G'_+.

(4.2) There is now a deficient vertex in a non-tree component H of G'_+.

It is easy to restore (3.1) in these cases. In case (4.1) holds, we alternately raise and lower z_j by ε on the edges of the path in T joining the deficient vertices, where ε is as large as possible. This either puts the two deficient vertices into different components of G'_+ or makes one of them non-deficient. In case (4.2) holds, the path in H from the deficient vertex to the odd cycle C contained in H gives an augmenting path of type 2 or 3. We perform the augmentation and shrink C if it becomes tight. This destroys the cycle, or puts the cycle and the deficient vertex in different components of G'_+, or makes the deficient vertex non-deficient. We remark that there is no guarantee that either of these operations will decrease the deficiency of z. The point is that they allow us to use Augment.

Now let us further suppose (as will be the case in the scaling algorithm) that each y_v and each $2z_j$ is an integral multiple of α. Then each augmentation found in re-optimizing will be of amount an integral multiple of α. But the deficiency (with respect to x') of z is at most 2α. Hence we can re-optimize by finding at most two augmentations of non-zero amount. So we can solve the problem for G and x', beginning with the solution for G and x, in time $O(|V|^2)$.

Now we can outline the scaling procedure. It applies to rational inputs, so let $x_v = p_v/q_v$, where for each v, q_v is a positive integer, p_v is a non-negative integer, and $p_v \leqslant q_v$. Let D denote the product (or the least common multiple) of the q_v, and let $r_v = Dx_v$ for $x \in V$. Let L denote $\lfloor \log(D+1) \rfloor$; all logarithms are in base 2. For $i = 0, 1, \ldots, L$, let $x^i = (x^i_v \colon v \in V)$ be defined by $x^i_v = r^i_v/D$, where $r^i_v = 2^i \lfloor r_v/2^i \rfloor$; that is, r^i_v is r_v with the i lowest order digits (base 2) replaced by zeros. *Problem i* is the separation problem for G and x^i. Hence Problem L is trivial, since $x^L = 0$, and Problem 0 is the problem we really want to solve, since $x^0 = x$. We solve problems $L, L-1, \ldots, 2, 1, 0$, in that order. Having solved problem $i+1$ for some i, $L > i \geqslant 0$, we solve problem i as follows. Let $V = \{v_1, v_2, \ldots, v_n\}$. For $j = 1, 2, \ldots, n$ replace x^{i+1}_{vj} by $x^i_{v_j}$ and apply the re-optimizing procedure. Notice that, if $x^i_{v_j} \neq x^{i+1}_{vj}$, then they differ by exactly $2^i/D$. But since all problems solved so far have each x^i_v an integral multiple of $2^i/D$, the re-optimizing procedure takes time $O(|V|^2)$ for each j. Hence Problem i is solved in time $O(|V|^3)$ and the original problem in time $O(|V|^3 L)$, which is polynomial in the input size.

Of course, it would be preferable to have an efficient algorithm whose running time does not depend so heavily on the length and form of the x_v. It seems likely one could obtain a strongly polynomial algorithm from the present one, using the approximation technique of Frank and Tardos (1987). However, a more natural strongly polynomial algorithm is desirable. For the case in which G is bipartite, such an algorithm is available, since the separation problem can be reduced to a minimum cut problem; see Ning (1987).

References

E. Balas and W.R. Pulleyblank, "The perfectly matchable subgraph polyhedron of an arbitrary graph," *Combinatorica* 9 (1989) 495–516.

W.H. Cunningham and J. Green-Krótki, "*b*-matching degree-sequence polyhedra," *Combinatorica* 11 (1991) 219–230.

J. Edmonds, "Maximum matching and a polyhedron with 0-1 vertices," *Journal of Research of the National Bureau of Standards* 69B (1965) 126–130.

J. Edmonds and E.L. Johnson, "Matching: A well-solved class of integer linear programs," in: R.K. Guy et al., eds. *Combinatorial Structures and their Applications* (Gordon and Breach, New York, 1970) pp. 89–92.

A. Frank and E. Tardos, "An application of simultaneous Diophantine approximation in combinatorial optimization." *Combinatorica* 7 (1987) 49–66.

M. Grötschel, L. Lovász and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica* 1 (1981) 169–197.

Q. Ning, "On separation and adjacency problems for perfectly matchable subgraph polytopes of a graph," *Operations Research Letters* 6 (1987) 239–242.

M.W. Padberg and M.R. Rao, "Odd minimum cuts and *b*-matchings," *Mathematics of Operations Research* 7 (1982) 67–80.

W.R. Pulleyblank, *Faces of Matching Polyhedra*, Ph.D. Thesis (University of Waterloo, Ont., Canada, 1973).