MATHEMATICS OF OPERATIONS RESEARCH
Vol. 7, No. 1, February 1982
Printed in U.S.A.

ODD MINIMUM CUT-SETS AND b-MATCHINGS*

MANFRED W. PADBERG† AND M. R. RAO‡

New York University

We show that the determination of a minimum cut-set of odd cardinality in a graph with even and odd vertices can be dealt with by a minor modification of the polynomially bounded algorithm of Gomory and Hu for multi-terminal networks. We connect this problem to the problem of identifying a matching (or blossom) constraint that chops off a point which is not contained in the convex hull of matchings or proving that no such inequality exists. Both the b-matching problems without and with upper bounds are considered. We discuss how the results of this paper can be used in conjunction with commercial LP packages to solve b-matching problems.

0. Introduction. The minimum cut-set problem for a finite undirected graph $G = (V, E)$ with nonnegative edge-weights c_e for all edges $e \in E$ consists of finding a partitioning of the nodes V of G into two nonempty subsets V_1 and V_2 such that the total weight of the edges connecting nodes in V_1 to nodes in V_2 is minimal. Gomory and Hu [11] have devised a polynomially bounded algorithm for the solution of this problem, see also [10] and [12]. Suppose now that the nodes of G have been partitioned into two classes of nodes which we call odd nodes and even nodes, respectively. A subset of nodes is called odd if it contains an odd number of odd nodes; even otherwise. The odd minimum cut-set problem is to find a partitioning of the nodes V of G into two (nonempty) *odd* subsets V_1 and V_2 such that the total weight of the edges connecting nodes in V_1 to nodes in V_2 is minimal. (We assume that G has an even number of odd nodes.)

In §1 we show that the odd minimum-cut set problem on such a "labelled" graph can be solved by a minor modification of the Gomory-Hu algorithm. In particular, the problem of finding an odd minimum cut-set can be solved in polynomial time. In §§2 and 3 we treat the b-matching problem without upper bounds and with upper bounds, respectively. The b-matching problem in a graph with positive integer node-numbers b_i for $i \in V$ and edge-weights $c_e \geqslant 0$ for $e \in E$ consists of finding a subset of edges such that every node is met by at most b_i edges and such that the total edge-weight of the selected edges is maximal. In §2 we treat the version where each edge can be picked an arbitrary (integer) number of times while in §3 we consider the version of the problem when an edge e can be picked at most d_e times where d_e is a positive integer number. The b-matching problem has been solved by Edmonds [4]–[6] who has devised a polynomially bounded algorithm for this class of problems. We show that the following problem can be solved by solving an odd minimum cut-set problem: Given a nonnegative vector $\bar{x}$ in the space of edge-variables, we want to decide whether or not $\bar{x}$ can be written as a convex combination of matchings (i.e., of integer points in edge-space which are feasible solutions to the b-matching problem). More precisely, we are interested in constructively providing a *cutting-plane* which will chop off $\bar{x}$ if it does

*Received August 23, 1979; revised September 2, 1980.
AMS 1980 subject classification. Primary: 05C35; Secondary: 90C10.
OR/MS Index 1978 subject classification. Primary: 486 Networks/graphs/matchings.
Key words. Odd minimum cut-sets, b-matching, cutting-planes, constraint identification, integer programming.
†Financial support under NSF Grant No. MCS 7908812.
‡On leave of absence from the Indian Institute of Management, Bangalore, India.

0364-765X/82/0701/0067$01.25

not belong to the convex hull of matchings. Naturally, we are not interested in finding *any* valid inequality or cutting-plane that does the job, but rather a member of the class of *matching constraints* (or blossom constraints) which have been shown by Edmonds [5] to define the convex hull of matchings; see also [1], [8]. We show that this *constraint identification problem* for b-matching problems—see also [2], [16] where the same question is addressed in the context of the travelling salesman problem—is an odd minimum cut-set problem on some suitably defined graph in both cases. Finally, in §4 we state an algorithmic application to the solution of b-matching problems which utilizes ideas from both primal and dual cutting-plane approaches to integer programming.

1. Odd minimum cut-sets. Let $G = (V, E)$ be a finite undirected graph without loops and multiple edges and let $c_e \geqslant 0$ for all $e \in E$ denote edge-weights. Let $V_1 \subseteq V$ be a nonempty set of nodes of G which are labelled odd. Let $V_0 = V - V_1$ be a (possibly empty) set of nodes which are labelled even. We extend the labelling of nodes to labels of subsets of nodes in the following natural way: For $W \subseteq V$ the (total) label $\lambda(W)$ is odd if W contains an odd number of nodes with odd label and the label $\lambda(W)$ is even otherwise. We assume that G has an even number of odd nodes, i.e., that the total label of V is even. We also use the convention that the empty set has an even label. A *cut-set* $(U : V - U)$ is called *odd* if $\lambda(U)$ is odd. The problem of finding an odd minimum cut-set in G can now be stated as follows: Find $W \subseteq V$ such that

$$c(W : V - W) = \min\{c(U : V - U) \mid U \subseteq V, \lambda(U) \text{ is odd}\} \tag{1.1}$$

where $c(W : V - W)$ is the capacity of the cut-set $(W : V - W)$, i.e.,

$$c(W : V - W) = \sum_{e \in (W : V - W)} c_e = \sum_{i \in W} \sum_{j \in V - W} c_{ij}.$$

As customary in the literature, we denote $e = [i, j]$ the undirected edge connecting node i and node j and drop the brackets and the comma when $[i, j]$ is used as an index. If W and U are disjoint node sets in V, the symbol $c(W : U)$ is used analogously with the convention that $c(W : U) = 0$ if W or U or $(W : U)$ is empty.

The next lemma shows that odd minimum cut-sets are related to (ordinary) minimum cut-sets in a very definite way. The implication is that problem (1.1) can be solved by a modification of the Gomory-Hu procedure for finding the minimum cut-set in a undirected graph [11], see also [10] and [12].

LEMMA 1.1. *Let $G = (V, E)$ and $c_e \geqslant 0$ for all $e \in E$ be given. Let $V = V_0 \cup V_1$ where V_1 is a nonempty set of nodes of G with odd label, V_0 is a set of nodes of G with even label and $\lambda(V)$ is even. Let $(M : V - M)$ be a minimum cut-set with respect to all pairs of odd labelled nodes in G. Then there exists an odd minimum cut-set $(X : V - X)$ in G such that $X \subseteq M$ or $X \subseteq V - M$ holds.*

PROOF. Let $(M : V - M)$ be a minimum cut-set with respect to all pairs of odd nodes in G. Suppose $\lambda(M)$ is even, for otherwise there is nothing to prove. Let $(T : V - T)$ be an odd minimum cut-set in G. For notational convenience, let $\overline{M} = V - M$ and $\overline{T} = V - T$. Suppose now that the assertion of the lemma is false. Then the four sets $M \cap T$, $M \cap \overline{T}$, $\overline{M} \cap T$ and $\overline{M} \cap \overline{T}$ must be nonempty since otherwise setting either $X = T$ or $X = \overline{T}$ would prove the lemma. Since $\lambda(M)$ is even and $\lambda(T)$ is odd, either $\lambda(M \cap T)$ or $\lambda(\overline{M} \cap T)$ is odd (but not both). Without any loss in generality, assume that $\lambda(M \cap T)$ is odd since otherwise M and $\overline{M}$ can be interchanged in the entire argument below. Now $\lambda(M \cap \overline{T})$ is also odd since $\lambda(M)$ is even. Since $\overline{M}$ contains at least two odd labelled nodes, either $\overline{M} \cap \overline{T}$ or $\overline{M} \cap T$ (or possibly both) must contain an odd node. We thus have two cases to consider:

Case 1. $\overline{M} \cap \overline{T}$ contains an odd node.

Now we show that $(M \cap T: V-(M \cap T))$ is an odd minimum cut-set. Let $W = M \cup (\overline{M} \cap T)$ and $\overline{W} = V - W$. Note that $W \neq M$ since $\overline{M} \cap T$ is nonempty. Since M as well as $\overline{M} \cap \overline{T}$ contains an odd node, both W and $\overline{W}$ contain an odd node. Now, minimality of $(M:\overline{M})$ implies that $c(M:\overline{M}) \leqslant c(W:\overline{W})$ or that

$$c(M \cap T: \overline{M} \cap T) + c(M \cap \overline{T}: \overline{M} \cap T) \leqslant c(\overline{M} \cap T: \overline{M} \cap \overline{T}).$$

Consequently, by the nonnegativity of the cut capacities, we have that

$$c(M \cap T: \overline{M} \cap T) \leqslant c(\overline{M} \cap T: \overline{M} \cap \overline{T}) + c(M \cap \overline{T}: \overline{M} \cap T). \tag{1.2}$$

Now, noting that

$$\begin{aligned} c(M \cap T: V-(M \cap T)) &= c(M \cap T: M \cap \overline{T}) + c(M \cap T: \overline{M} \cap T) \\ &\quad + c(M \cap T: \overline{M} \cap \overline{T}), \\ c(T:\overline{T}) = c(M \cap T: M \cap \overline{T}) &+ c(M \cap T: \overline{M} \cap \overline{T}) + c(\overline{M} \cap T: M \cap \overline{T}) \\ &+ c(\overline{M} \cap T: \overline{M} \cap \overline{T}), \end{aligned}$$

from (1.2) we have that

$$c(M \cap T: V-(M \cap T)) \leqslant c(T:\overline{T}).$$

Consequently, the lemma follows if we set $X = M \cap T$.

Case 2. $\overline{M} \cap T$ contains an odd node.

Now interchanging T and $\overline{T}$ in Case 1 above, we have that $(M \cap \overline{T}: V-(M \cap \overline{T}))$ is an odd minimum cut-set. Consequently the lemma follows with $X = M \cap \overline{T}$.

In the terminology of Hu [12, p. 132] it follows from Lemma 1.1 that in a labelled graph for every cut-set which is minimum with respect to all pairs of odd labelled nodes there exists a *noncrossing* odd minimum cut-set in G.

Lemma 1.1 implies that the odd minimum cut-set problem can be solved by working on condensed or shrunk graphs; in other words, the lemma implies that the main aspect of the Gomory-Hu procedure remains intact here, too. For let $(M: V-M)$ be a minimum cut-set with respect to all pairs of odd nodes of G. If $\lambda(M)$ is odd, we are done. Else we define two new graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$ where $V^1 = M \cup \{s^1\}$ and $V^2 = (V-M) \cup \{m^1\}$; s^1 is the new node obtained by shrinking $V-M$ into a single node and m^1 is a new node obtained by shrinking M into a single node. An edge and its edge-weights in the new graph G^i, $i = 1$ or 2, is an original edge and its respective weight, if it connects two "original" nodes; otherwise, let node i which is not in the shrunk part of the new graph be connected to nodes $j_1, \ldots, j_k$ all of which are in the shrunk part of the new graph.

Then node i is connected to the new node by a single edge with weight $c_{ij_1} + \cdots + c_{ij_k}$. This way of shrinking the graph G is the precise same construction that Gomory and Hu [11] use. We observe next that the new node obtained by shrinking M or $V-M$ "inherits" an even label since both $\lambda(M)$ and $\lambda(V-M)$ are even. Furthermore, both $\lambda(V^1)$ and $\lambda(V^2)$ are even. It follows, e.g., from Lemma 3.1 in [10, p. 179] that a minimum cut-set with respect to all pairs of odd nodes in G^1 (in G^2, respectively) is a minimum cut-set in G with respect to all pairs of odd nodes in M (in $V-M$, respectively). Furthermore, since $(M: V-M)$ by construction separates two odd nodes in G, M ($V-M$, respectively) contains at least two odd nodes. Thus G^i for $i = 1, 2$ contains at least two odd nodes and at least two odd nodes less than G did. We can thus solve the minimum cut-set problem in G^i for $i = 1$ and 2 and repeat as Lemma 1.1 applies to G^i for $i = 1, 2$. After at most as many steps as there are odd nodes in G we find an odd minimum cut-set by choosing a smallest one among the odd minimum cut-sets obtained during the entire calculation.

The above discussion shows that problem (1.1) can be solved by a successive calculation of minimum cut-sets in condensed or shrunk graphs. Since, however, the set of odd nodes of any shrunk graph is a proper subset of the odd nodes of its "ancestor" graphs, it follows from the results of Gomory-Hu [11] that the entire calculation can be *streamlined considerably* and that it is necessary to carry out the minimum cut-set calculation *only once* for all pairs of odd nodes in the graph. Let $G_T = (N, F)$ be the *cut-tree* obtained by applying the Gomory-Hu algorithm to all pairs of odd labelled nodes of G. (The odd nodes are thus "terminal" nodes as defined by Hu [12, pp. 131 ff.].) N is the node-set of the cut-tree; every node in N corresponds to one or several nodes of G and contains exactly one odd labelled node of G. F contains all the edges of the cut-tree and we denote by d_f the (nonnegative) weight for $f \in F$. Let f connect two nodes of G_T containing odd node i_1 and odd node i_2 of the original graph G. Then d_f is the capacity of a minimum cut-set of G separating nodes i_1 and i_2 (see Theorem 9.2 in [12, p. 138]). The minimum of the values d_f for all $f \in F$ constitutes the capacity of a minimum cut-set separating any two odd nodes in G.

Let $f^* \in F$ be an edge for which the minimum is attained, i.e. $d_{f^*} = \min\{d_f / f \in F\}$. Removing f^* from the tree G_T creates two new trees G_T^1 and G_T^2 with node-sets N_1 and N_2. In terms of the original graph the nodes in N_1 correspond to a set M and the nodes in N_2 to the complement $V - M$ in G such that $(M : V - M)$ is a minimum cut-set with respect to all pairs of odd nodes in G. If the cut-set is odd, we have solved problem (1.1). If not, we can shrink as discussed previously and form the graphs G^1 and G^2. But G_T^1 is a cut-tree for G^1 and G_T^2 is a cut-tree for G^2, since the shrinking does not create any new odd nodes and the unique path of G_T from an odd node i_1 of G^1 to some other odd node i_2 of G^1 is contained in G_T^1 and likewise for G^2. Consequently by Theorem 9.2 in [12, p. 138] we need only determine a minimum weight edge in G_T^1 to find a minimum cut-set in G^1 and likewise, a minimum weight edge in G_T^2 in order to find a minimum cut-set in G^2. Thus in order to locate an odd minimum cut-set in G we have to find G_T and then "cut up" the cut-tree until we find an edge whose removal leaves two subtrees which have both an odd number of nodes. Of course, this cutting up has to be performed for every subtree that is generated in the process.

Let $f_1, \ldots, f_p$ be the edges that are found eventually. By construction, removal of any of those edges creates two subtrees in G_T which both have an odd number of nodes thereby yielding an odd cut-set in G. An odd minimum cut-set in G is obtained by choosing the minimum among $f_1, \ldots, f_p$. Summarizing, we have proven the following theorem:

THEOREM 1.1. *Let G be a graph whose nodes have been labelled odd and even such that the total label of G is even. An odd minimum cut-set is defined by a minimum weight edge of the cut-tree G_T whose removal decomposes G_T into two subtrees of odd cardinality.*

By cardinality of a subtree we mean the number of nodes in the subtree. Furthermore from the properties of the cut-tree G_T obtained by applying the Gomory-Hu algorithm to all pairs of odd nodes of G, it follows that the odd minimum cut-set thus obtained is a (ordinary) minimum cut-set separating two odd nodes of G.

The following example illustrates the method of determining an odd minimum cut-set in a labelled graph. It also shows that an analog of Theorem 1.1 is false if one wants to find an odd minimum cut-set separating two *given* odd nodes. This latter problem is apparently harder to solve than the one we are addressing here. (To see that the example provides a counterexample determine an odd minimum cut-set separating nodes 2 and 5 in the example below.)

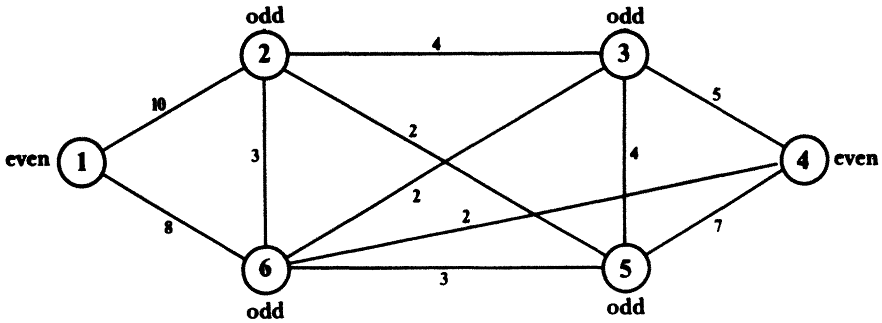
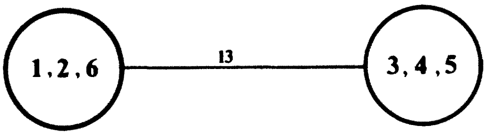
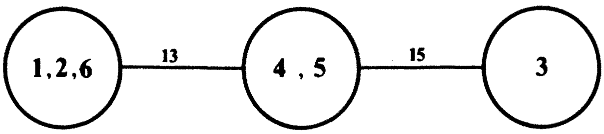


FIGURE 1

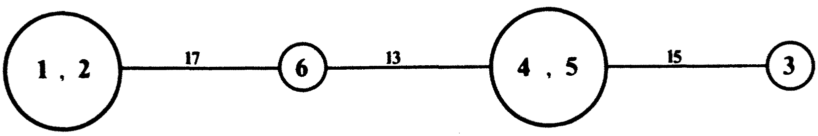
EXAMPLE. The following example is taken from Gomory-Hu [11] except for the labelling. The nodes 2, 3, 5, 6 are labelled odd, while nodes 1 and 4 are labelled even. We start by selecting (arbitrarily) nodes 2 and 5; a minimum cut-set separating 2 and 5 can be read off of the cut-tree determined in [11] and is given by $(\{1, 2, 6\} : \{3, 4, 5\})$ with capacity 13.



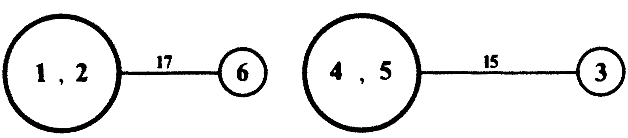
Now we select nodes 3 and 5 and treating $\{1, 2, 6\}$ as a single node we find a minimum cut-set separating 3 and 5 as given by $(\{\{1, 2, 6\}, 4, 5\} : \{3\})$ with capacity 15.



Finally, we select 2 and 6 and treating $\{3, 4, 5\}$ as a single node we find a minimum cut-set to be $(\{1, 2\} : \{6, \{3, 4, 5\}\})$ with capacity 17. Thus the cut-tree of all pairs of odd nodes of the example graph is given by



The minimum weight in the cut-tree is 13 and removing the corresponding edge creates the two subtrees



The labels of the corresponding subsets are both even and thus we cut up the tree further. Cutting up the left tree yields

and thus an odd cut-set with capacity 17 is obtained and given by $(\{1, 2\} : \{3, 4, 5, 6\})$. Cutting up the right tree we get

and thus an odd-cut with capacity 15 is obtained and given by $(\{1, 7, 4, 5, 6\} : \{3\})$. Since $15 < 17$, the latter solves the odd min-cut problem on the example graph.

2. Constraint identification in *b*-matching problems. The b-matching problem is the following combinatorial optimization problem: Maximize cx subject to

$$Ax \leqslant b, \qquad x \geqslant 0 \tag{2.1}$$

and x integer, where c is a (nonnegative) vector of m reals, A is the $n \times m$ incidence matrix of an undirected graph having n nodes and m edges and b is a vector of n positive integer numbers. The problem has been solved by Edmonds [4], [5] who has given a polynomially bounded algorithm for this class of problems. The b-matching problem is a genuine integer problem, i.e., not all basic feasible solutions to (2.1) are automatically integer. Rather, additional linear inequalities are required to obtain a constraint system whose basic feasible solutions are all integer. This additional class of inequalities are called *matching constraints* (or blossom constraints) and have the following general form: Let $G = (V, E)$ be the undirected graph whose incidence matrix is given by A and let $W \subseteq V$ be any subset of nodes of G (i.e., subsets of rows of A) such that

$$b(W) = \sum_{i \in W} b_i \tag{2.2}$$

is odd. Then the inequality

$$x(W) = \sum_{e \in E(W)} x_e \leqslant \tfrac{1}{2}(b(W) - 1) \tag{2.3}$$

is a valid inequality for the integer solutions to (2.1), where $E(W) \subseteq E$ is the set of edges of G having both ends in W. Edmonds has shown—by giving a good, i.e., polynomially bounded, algorithm for the matching problem—that the totality of all constraints (2.1) and (2.3) defines a polyhedron (the b-matching polyhedron) whose extreme points are all integer. Pulleyblank [18] has characterized the (unique) minimal system of linear inequalities (2.1) and (2.3) that define the matching polyhedron.

The question that interests us is the following:

(P1) *Let $\bar{x}$ satisfy* (2.1) *and suppose that $\bar{x}$ is not integral. Find a subset $W \subseteq V$ such that $b(W)$ is odd and such that $\bar{x}(W) > \frac{1}{2}(b(W) - 1)$ holds or prove that no such $W \subseteq V$ exists.*

The problem (P1) is encountered, e.g., when one uses a (dual) cutting-plane algorithm for the b-matching problem and wishes to employ standard software packages for linear programming problems (such as IBM's MPSX or CDC's APEX packages) for the solution of b-matching problems. As it turns out, problem (P1) can be solved by solving an odd minimum cut-set problem on a suitably defined graph.

We first write the inequality (2.3) in a different form. Let s be the vector of slack variables for the constraints $Ax \leqslant b$. Denoting

$$s(W) = \sum_{i \in W} s_i \tag{2.4}$$

we can then write inequality (2.3) in the equivalent form

$$x(W : V - W) + s(W) \geqslant 1. \tag{2.5}$$

This follows since

$$2x(W) + x(W : V - W) + s(W) = b(W) \tag{2.6}$$

holds by adding up the rows of $Ax + s = b$ with index in W. Using (2.3) one then obtains (2.5). (Inequality (2.3) follows from (2.6) by dividing by two and integerization.) Given $\bar{x}$ define $\bar{s} = b - A\bar{x}$ to be the vector of slack variables. Then $\bar{x}(W) > \frac{1}{2}(b(W) - 1)$ holds if and only if

$$\bar{x}(W : V - W) + \bar{s}(W) < 1 \tag{2.7}$$

holds. Let $G(\bar{x})$ be the graph with node-set $V^* = V \cup \{S\}$ and edge-set $E(\bar{x})$. The node S denotes a special node to account for the slack variables. The graph $G(\bar{x})$ has an edge e with edge-weight $\bar{x}_e$ iff e connects two nodes of V and $\bar{x}_e > 0$ holds. In addition, $G(\bar{x})$ has edges with edge-weight $\bar{s}_i$ between a node i in V and the special node S iff $\bar{s}_i > 0$ holds. We label a node $i \in V$ odd iff b_i is odd; $i \in V$ is labelled even otherwise. If the total label $\lambda(V)$ of V is odd, we label S odd; S is labelled even otherwise. The problem (P1) is thus to find an odd minimum cut-set in $G(\bar{x})$. If the capacity of an odd minimum cut-set in $G(\bar{x})$ is greater than or equal to 1, we conclude that $\bar{x}$ is in the convex hull of integer solutions to the b-matching problem, since it cannot be chopped off by a matching constraint. Else, the procedure of §1 yields an odd minimum cut-set $(W : V^* - W)$ with $S \in V^* - W$ and satisfying (2.7), i.e., a matching constraint is identified that is *most violated* by the feasible solution $\bar{x}$ to (2.1). In addition, the calculation using the procedure of §1 will generally identify several other matching constraints that are violated by $\bar{x}$. Furthermore, since the effort spent on calculating an odd minimum cut-set is bounded by a polynomial in the number of nodes of the underlying graph G, the identification of a matching constraint can be done in polynomial time. This fact lends substantive meaning to the statement that matching constraints are *combinatorially pleasant*; see [7, p. 132].

Due to the fact that in answering question (P1) we are interested in finding an odd minimum cut-set with a cut capacity less than 1 one can speed up the calculations considerably by shrinking the graph $G(\bar{x})$ prior to actually determining an odd minimum cut-set. For let $e \in E(\bar{x})$ be an edge with edge-weight $\bar{x}_e \geqslant 1$ or $\bar{s}_i \geqslant 1$ and let k and j be the two nodes connected by this edge. Since all edge-weights are positive, the cut capacity is greater than or equal to the edge-weight of any particular edge in the cut-set. Consequently, any odd minimum cut-set of capacity less than 1 cannot contain any edge with edge-weight greater than or equal to 1. Thus nodes k and j can be shrunk into a (pseudo-) node. The new node is labelled odd if exactly one of the nodes k and j is odd; even otherwise. The new node is connected to all neighbors of k and j with the original edge-weights; if k and j have a common neighbor a new edge-weight is defined as the *sum* of the two old edge-weights. Iterating this procedure we arrive at a (generally considerably smaller) graph having all edge-weights strictly less than one. If the resulting graph has only even nodes, we conclude that there exists no matching constraint which chops off $\bar{x}$. Alternatively, we carry out an odd minimum cut-set calculation on the shrunk graph and "expand" the corresponding odd cut-sets to cut-sets in $G(\bar{x})$.

A further reduction in the calculation is achieved by considering connected components of $G(\bar{x})$. Since $\bar{x}$ is an arbitrary feasible solution to (2.1) it may very well happen that $G(\bar{x})$ decomposes into several connected components. (Note that in §1 we did not require that the graph be connected.) Suppose that $G(\bar{x})$ decomposes into q connected components $G_1, \ldots, G_q$. If any one of the connected components not containing the special node S has a total label which is odd, the node set of such component defines a node set W such that $(W: V^* - W)$ is an odd minimum cut-set since its cut capacity is zero. On the other hand, if the connected component containing the special node S has a total label which is odd, then there must exist some other connected component of $G(\bar{x})$ which has a total label which is odd since the total label of the node-set V^* of $G(\bar{x})$ is even. Thus we can assume that every connected component of $G(\bar{x})$ has an even label. Let $(W: V^* - W)$ be an odd cut-set in $G(\bar{x})$ where $S \notin W$ and $\lambda(W)$ is odd. It follows that W intersects an odd number of the connected components G_i such that $\lambda(W \cap G_i)$ is odd. If $(W: V^* - W)$ satisfies (2.7) then each $W_i = W \cap G_i$ with $\lambda(W_i)$ odd yields a cut-set $(W_i: V^* - W_i)$ which also satisfies (2.7) and thus the calculation can be restricted to the connected components of $G(\bar{x})$.

A final word concerns the case where all inequalities or some of the inequalities of (2.1) are required to hold as equations. In this case the analysis proceeds as before except that now we have only equations (perfect b-matching), so there is no need for a special node S or the introduction of slack variables.

3. Constraint identification in b-matching problem with upper bounds. The b-matching problem with upper bounds is the following combinatorial optimization problem: Maximize cx subject to

$$\begin{aligned} Ax &\leqslant b, \\ x &\leqslant d, \\ x &\geqslant 0 \end{aligned} \tag{3.1}$$

and x integer, where c is a (nonnegative) vector of m reals, A is the $n \times m$ incidence matrix of an undirected graph having n nodes and m edges, b is a vector of n positive integer numbers and d is a vector of m positive integers. This problem has been treated in the more general context of bi-directed networks by Edmonds and Johnson [8], [13], see also [9] for a computer code implementation which gives a polynomially bounded computational procedure for b-matching problems with upper bounds. Like the b-matching discussed in §2, the problem at hand is a genuine integer programming problem. The additional constraints needed to linearly describe the convex hull of integer solutions to (3.1) are again called *matching constraints* (or blossom constraints) and have the following general form: Let $G = (V, E)$ be the undirected graph whose incidence matrix is given by A. Let $W \subseteq V$ be any subset of nodes of G and let $T \subseteq (W: V - W)$ be any subset of edges of G which are in the cut-set furnished by W. (If $W = V$, T is taken to be empty.) If the quantity

$$b(W) + d(T) = \sum_{i \in W} b_i + \sum_{e \in T} d_e \tag{3.2}$$

is odd, then the inequality

$$x(W) + x(T) = \sum_{e \in E(W)} x_e + \sum_{e \in T} x_e \leqslant \tfrac{1}{2}(b(W) + d(T) - 1) \tag{3.3}$$

is a valid inequality for the integer solutions to (3.1), where for any node-set W the symbol $E(W)$ denotes the set of edges of G with both ends in W. It has been proven algorithmically by Edmonds that the totality of all constraints (3.1) and (3.3) defines a polyhedron whose extreme points are all integer.

The question that interests us is analogous to question (P1) in §2:

(Q1) *Let* $\bar{x}$ *satisfy* (3.1) *and suppose that* $\bar{x}$ *is not integral. Find subsets* $W \subseteq V$ *and* $T \subseteq (W : V - W)$ *such that* $b(W) + d(T)$ *is odd and such that* $\bar{x}(W) + \bar{x}(T) > \frac{1}{2}(b(W) + d(T) - 1)$ *holds or prove that no such* W *and* T *exist.*

The motivation for this problem is the same as for (P1) as we would like to use standard software packages for linear programming problems in the solution of the b-matching problems with upper bounds.

We first write the inequality (3.3) in a different form. Let s be the vector of slack variables for the constraints $Ax \leqslant b$ and t be the vector of slack variables for the upper bound constraints. Denoting $s(W)$ as previously (see (2.4)) and

$$t(T) = \sum_{e \in T} (d_e - x_e) = \sum_{e \in T} t_e \tag{3.4}$$

we can write inequality (3.3) in the equivalent form

$$\sum_{i \in W} \sum_{\substack{j \in V - W \\ [i,j] \notin T}} x_{ij} + \sum_{e \in T} (d_e - x_e) + \sum_{i \in W} s_i \geqslant 1. \tag{3.5}$$

This follows since

$$2x(W) + x(W : V - W) + x(T) + s(W) + t(T) = b(W) + d(T) \tag{3.6}$$

holds by adding up the rows of $Ax + s = b$ with index in W plus the rows of $x + t = d$ with index in T. Using (3.3) one then obtains (3.5) after substituting $t(T) = d(T) - x(T)$. (Inequality (3.3) follows from (3.6) by dividing by two and integerization.) Given $\bar{x}$ define $\bar{s} = b - A\bar{x}$ and $\bar{t} = d - \bar{x}$ to be the vector of respective slack variables. Then $\bar{x}$ violates inequality (3.5) iff

$$\bar{x}(W : V - W) + d(T) - 2\bar{x}(T) + \bar{s}(W) < 1 \tag{3.7}$$

holds. Note that in inequality (3.7) (as well as in (3.5)) all edges in $(W : V - W) - T$ occur with the "correct" edge-value, while the edge-values of all edges in T appear in "complemented" form. Thus inequality (3.7) is again a cut-set constraint of some kind, see also [8], [13]. We want to show next that the problem (Q1) again leads to the problem of finding an odd minimum cut-set in a suitably defined graph.

Given $\bar{x}$ and $\bar{s}$ we define the graph $G(\bar{x})$ with node-set $V^* = V \cup \{S\}$ and edge-set $E(\bar{x})$ as done in §2. Also, we label the nodes of V as done previously. We next construct a new graph $G(\bar{x}, d)$ by the following procedure: We scan through the edges of $G(\bar{x})$ one-by-one. Let $e = [i, j]$ be the current (unscanned) edge of $G(\bar{x})$. If $i = S$ or $j = S$, take the next edge. Else, replace edge e by two edges $[i, i_e]$ and $[i_e, j]$ where i_e is a new node. Node i_e is labelled odd if d_e is odd; even otherwise. Node i gets an even label, if its old label and the label of i_e are identical; odd otherwise. Node j retains its old label. The edge $[i, i_e]$ gets the edge-weight $d_e - \bar{x}_e$ and the edge $[i_e, j]$ gets the edge-weight $\bar{x}_e$. Then we choose the next edge of $G(\bar{x})$, use the redefined labels and iterate until all edges of $G(\bar{x})$ have been scanned.

Denote $\tilde{V} = V^* \cup V_*$ where $V_* = \{i_e \mid e \in E(\bar{x})\}$ is the set of newly introduced nodes. Note that $\lambda(\tilde{V})$ is even since $\lambda(V^*)$ is even. (This follows inductively from the assignment of new labels.) It follows from the construction that for every edge $e \in E(\bar{x})$ there exists a uniquely defined new node i_e in $G(\bar{x}, d)$. Denote F the edge-set of $G(\bar{x}, d)$ and let $\bar{y}_f$ for $f \in F$ be the edge-weight of edge f.

LEMMA 3.1. *Let $W \subseteq V$ and $T \subseteq (W : V - W) \cap E(\bar{x})$ be such that $b(W) + d(T)$ is odd. Then there exist an odd cut-set $(U : \tilde{V} - U)$ in $G(\bar{x}, d)$ with $S \in \tilde{V} - U$ and such that*

$$\bar{x}(W : V - W) + d(T) - 2\bar{x}(T) + \bar{s}(W) = \bar{y}(U : \tilde{V} - U) \tag{3.8}$$

holds.

PROOF. Let W_1 and U_1 be defined as follows

$$\begin{aligned} W_1 &= \{ i_e \in V_* \mid e \in E(W) \cap E(\bar{x}) \}, \\ U_1 &= \{ i_e \in V_* \mid e \in (W : V - W) \cap E(\bar{x}) \}. \end{aligned} \tag{3.9}$$

We partition U_1 into four disjoint sets as follows:

$$\begin{aligned} U_{11} &= \{ i_e \in U_1 \mid e \notin T, \bar{y}_{ki_e} = d_e - \bar{x}_e \text{ for some } k \in W \}, \\ U_{12} &= \{ i_e \in U_1 \mid e \notin T, \bar{y}_{ki_e} = \bar{x}_e \text{ for some } k \in W \}, \\ U_{13} &= \{ i_e \in U_1 \mid e \in T, \bar{y}_{ki_e} = d_e - \bar{x}_e \text{ for some } k \in W \}, \\ U_{14} &= \{ i_e \in U_1 \mid e \in T, \bar{y}_{ki_e} = \bar{x}_e \text{ for some } k \in W \}. \end{aligned} \tag{3.10}$$

Define $U = W \cup W_1 \cup U_{11} \cup U_{14}$. Since $S \notin U$ it follows by the construction of the graph $G(\bar{x}, d)$ and the assignment of the labels of the nodes of $G(\bar{x}, d)$ that the label of U is determined by

$$\sum_{i \in W} b_i + 2 \sum_{e \in E(W) \cap E(\bar{x})} d_e + 2 \sum_{\{e \mid i_e \in U_{11}\}} d_e + \sum_{\{e \mid i_e \in U_{14}\}} d_e + \sum_{\{e \mid i_e \in U_{13}\}} d_e. \tag{3.11}$$

Since $b(W) + d(T)$ is odd, it follows that the label of U is odd. From the construction of $G(\bar{x}, d)$, since $(W_1 : \tilde{V} - U)$ is empty it follows further that

$$\begin{aligned} \bar{y}(U : \tilde{V} - U) &= \bar{y}(W : \tilde{V} - U) + \bar{y}(U_{11} : \tilde{V} - U) + \bar{y}(U_{14} : \tilde{V} - U) \\ &= \bar{s}(W) + \sum_{\{e \mid i_e \in U_{12}\}} \bar{x}_e + \sum_{\{e \mid i_e \in U_{11}\}} \bar{x}_e + \sum_{\{e \mid i_e \in U_{13}\}} (d_e - \bar{x}_e) + \sum_{\{e \mid i_e \in U_{14}\}} (d_e - \bar{x}_e) \\ &= \bar{s}(W) + \bar{x}((W : V - W) \cap E(\bar{x}) - T) + d(T) - \bar{x}(T) \end{aligned} \tag{3.12}$$

holds. Consequently (3.8) follows.

LEMMA 3.2. *Let $(U : \tilde{V} - U)$ be an odd cut-set in $G(\bar{x}, d)$ with cut-capacity $c < 1$ and such that $S \notin U$. Then there exist subsets $W \subseteq V$ and $T \subseteq (W : V - W)$ such that $b(W) + d(T)$ is odd and such that* (3.8) *holds.*

PROOF. Define $W = U \cap V$ and let W_1 be defined as in (3.9). Denote $W_2 = U \cap V_* - W_1$. If $i_e \in W_2$, let i and j be the nodes of V determined by $e \in E(\bar{x})$. If both $i \notin W$ and $j \notin W$, then by construction of the graph $G(\bar{x}, d)$ we have $c \geqslant \bar{x}_e + d_e - \bar{x}_e = d_e \geqslant 1$, a contradiction to our assumption that $c < 1$. Consequently, since $U \neq \emptyset$, it follows that $W \neq \emptyset$. On the other hand, let $e \in E(W) \cap E(\bar{x})$, $e = [i, j]$ and suppose that $i \in W$, but $j \in W$, but $i_e \notin U$. Again it follows by the construction of $G(\bar{x}, d)$ that $c \geqslant \bar{x}_e + d_e - \bar{x}_e \geqslant 1$, contradicting $c < 1$. Consequently, $W_1 \subseteq U$ and there exist no edges connecting nodes in W_1 to nodes in $\tilde{V} - U$. Hence by construction *all* edges in $(U : \tilde{V} - U)$ connect nodes in W to some node in $V_* \cup \{S\}$ or some node in W_2 to some node in $V - W$. Define T as follows:

$$\begin{aligned} T = \{ e \in E(\bar{x}) \mid \exists f \in (U : \tilde{V} - U) & \text{ such that} \\ & f = [i_e, j] \text{ for some } j \in V \text{ and } \bar{y}_f = d_e - \bar{x}_e \}. \end{aligned} \tag{3.13}$$

Define U_1 as done previously in (3.9). It follows that the choice of T given by (3.13) partitions U_1 into four disjoint sets as in (3.10) and furthermore, that $U = W \cup W_1 \cup U_{11} \cup U_{14}$ holds. Thus the calculation of the label of U (3.11) remains valid and since it is odd, it follows that $b(W) + d(T)$ is odd. The calculation of $\bar{y}(U : \tilde{V} - U)$ in (3.12) remains valid as well. Consequently, (3.8) holds and the lemma follows.

The two lemmata furnish the desired result:

THEOREM 3.1. *Given a feasible solution $\bar{x}$ to* (3.1) *there exist $W \subseteq V$ and $T \subseteq (W : V - W)$ such that $b(W) + d(T)$ is odd and such that $\bar{x}(W) + \bar{x}(T) > \frac{1}{2}(b(W) + d(T) - 1)$ holds if and only if the cut-capacity of an odd minimum cut-set in $G(\bar{x}, d)$ is less than one. Furthermore, W and T are obtained constructively as in the proof of Lemma* 3.2.

PROOF. If W and T exist with the required properties, then it follows by Lemma 3.1 that the cut-capacity of an odd minimum cut-set in $G(\bar{x}, d)$ is less than one because (3.7) is equivalent to $\bar{x}(W) + \bar{x}(T) > \frac{1}{2}(b(W) + d(T) - 1)$. On the other hand, if the capacity of an odd minimum cut-set in $G(\bar{x}, d)$ is less than one then W and T exist and have the desired properties by Lemma 3.2. The proof of Lemma 3.2 shows how to construct W and T from an odd minimum cut-set in $G(\bar{x}, d)$.

To decide question (Q1) we thus have to solve an odd minimum cut-set problem in the labelled graph $G(\bar{x}, d)$ and all of the techniques discussed in Section 2 can be used here as well to speed up the calculation of an odd minimum cut-set. Thus prior to applying the Gomory-Hu algorithm we can shrink all edges with edge-weights greater than or equal to one. We check the label of all connected components and then determine an odd minimum cut-set in the (usually considerably smaller) shrunk graph. Because of Theorem 3.1 it follows that there exists a matching constraint (3.3) which chops off the feasible solution $\bar{x}$ to (3.1) if and only if the cut capacity of an odd minimum cut-set in $G(\bar{x}, d)$ is less than one. Furthermore the pair W and T is obtained constructively. If the odd minimum cut-capacity is greater than or equal to one, $\bar{x}$ is a convex combination of integer feasible solutions to (3.1). Furthermore, while the determination of a *minimum* odd cut-set in $G(\bar{x}, d)$ finds a *most violated* matching constraint in polynomial time, the results in this section show that *all* odd cut-sets in $G(\bar{x}, d)$ with capacity *less than one* furnish violated matching constraints.

For matters of completeness we note that, like in the case discussed in §2, some or all of the inequalities in (3.1) may actually be equations. In the latter case there is no need to consider the introduction of a special node. Finally, while in treating the b-matching problem with upper bounds we have assumed that upper bounds d_e are specified for *all* edges $e \in E$, it is clear how to accomodate the situation where there are no explicit upper bounds on some of the variables. In this case the construction of the graph $G(\bar{x}, d)$ is modified slightly since there is no need to define a new node i_e for such unrestricted edges, i.e., such edges are simply skipped in setting up $G(\bar{x}, d)$. Everything else, including the constructive characterization of W and T in the proof of Lemma 3.2, remain the same.

4. Concluding remarks. Most commercial software packages for solving linear programming problems, such as IBM's MPSX/370 or CDC's APEX package, permit the user to "revise" the linear programming problem. Such revision of the original problem includes the possibility of appending new constraints to a linear programming problem. These program products are often the result of several man-years of effort by highly skilled mathematicians, computer scientists and computer programmers. It is therefore natural to look for ways to utilize such highly developed commercial codes as "building blocks" for the solution of combinatorial optimization problems. In [2] an application of this idea to the solution of symmetric travelling salesman problems has

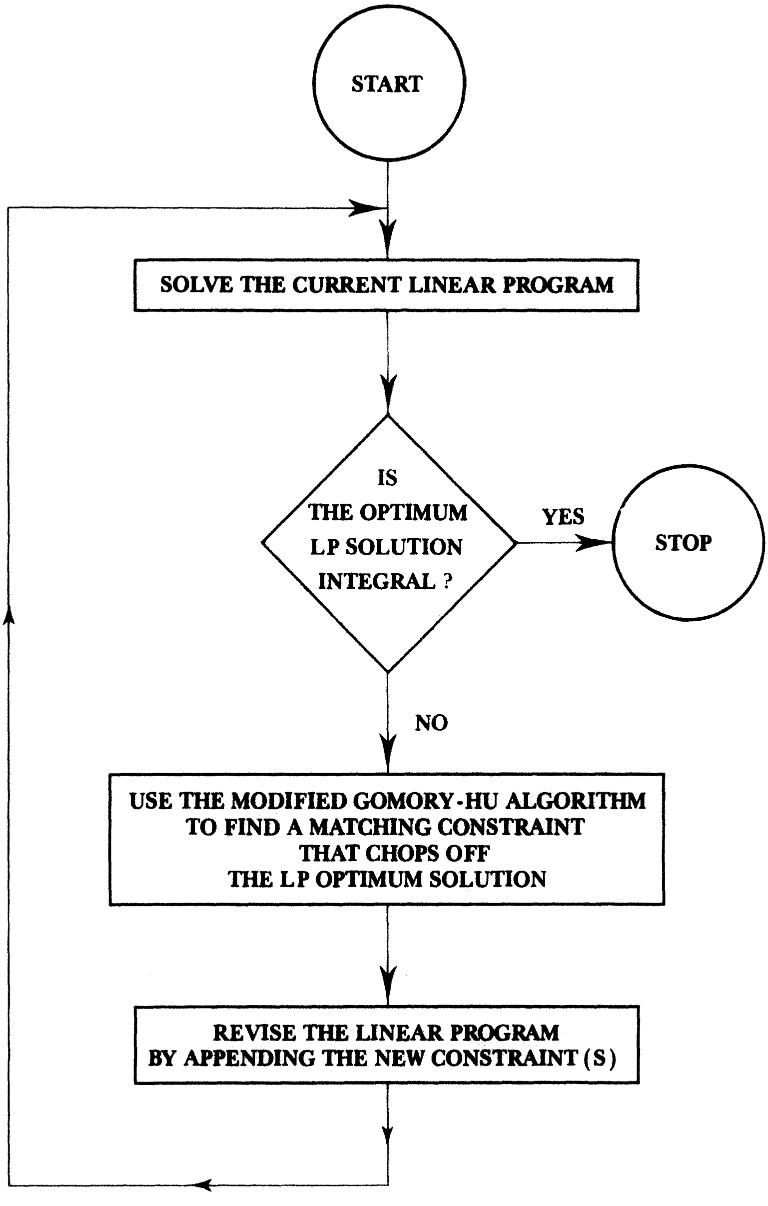


FIGURE 2. Flowchart.

been described. The results of the present paper permit to use existing commercial software in the solution of b-matching problems without resorting to branch-and-bound.

In Figure 2 we give a schematic flow-chart for the calculation. Of course, the basic idea is to use cutting-planes and this idea is not exactly new. What is different are the cutting-planes (matching constraints) employed in the calculation and, from a computational point of view, this makes a great difference. While the algorithm for b-matching problems that we get that way is not polynomially bounded, it is quite reasonable to expect satisfactory computational performance because the cutting-planes are mathematically proven to be good cutting-planes since their totality defines the convex hull of matchings. Because this is the case and because there are only finitely many matching constraints, the finiteness of the computational procedure described in Figure 2 follows from the finiteness of the simplex method (or any other finite linear programming algorithm employed in solving the sequence of linear programs).

Besides the problems (P1) and (Q1) which embody the basic idea of a *dual* cutting-plane method for integer programming problems, it is of interest to note that the idea embodied into a *primal* cutting plane method can be dealt with entirely analogously. See [3] for a different primal algorithm for this class of problems. The corresponding problem for b-matching problems is the following:

(P2) *Let x^1 be an integral solution satisfying* (2.1) *and suppose that x^2 satisfies* (2.1), *but is not integral. Find a subset $W \subseteq V$ such that $b(W)$ is odd and such that $x^1(W) = \frac{1}{2}(b(W) - 1)$ and $x^2(W) > \frac{1}{2}(b(W) - 1)$ hold or prove that no such $W \subseteq V$ exists.*

To answer the problem define the graph $G(\bar{x})$ of Section 2 with $\bar{x} = (x^1 + x^2)/2$. Then every odd cut-set in $G(\bar{x})$ has a capacity of one half or more since (2.5) is satisfied by x^1. Consequently, a violated matching constraint which is satisfied by x^1 at equality and chops off x^2 corresponds to an odd cut-set with cut-capacity less than one. The reverse follows since for every integer feasible solution to (2.1) the left hand-side of (2.5) is an integer number no matter what $W \subseteq V$. Consequently, (P2) can be answered constructively by calculating an odd minimum cut-set in the graph $G(\frac{1}{2}x^1 + \frac{1}{2}x^2)$.

The corresponding primal question for b-matching problems with upper bounds is the following:

(Q2) *Let x^1 be an integral solution satisfying* (3.1) *and suppose that x^2 satisfies* (3.1), *but is not integral. Find subsets $W \subseteq V$ and $T \subseteq (W : V - W)$ such that $b(W) + d(T)$ is odd and such that $x^1(W) + x^1(T) = \frac{1}{2}(b(W) + d(T) - 1)$ and $x^2(W) + x^2(T) > \frac{1}{2}(b(W) + d(T) - 1)$ hold or prove that no such W and T exist.*

To answer (Q2) we define $G(\bar{x})$ of §2 with $\bar{x} = (x^1 + x^2)/2$ and construct $G(\bar{x}, d)$ as done in §3. Since x^1 satisfies (3.5) for every permissible choice of W and T and yields an integer number for the left-hand side of (3.5), it follows from Lemma 3.1 and Lemma 3.2 that an odd minimum cut-set in $G(\bar{x}, d)$ with cut-capacity less than one yields a most violated matching constraint that is satisfied by x^1 at equality and chops off x^2. The reverse statement holds as well.

The odd minimum cut-set calculation can thus be employed to devise both primal and dual cutting-plane algorithms for b-matching problem with and without upper bounds. When this paper was presented at the X International Symposium on Mathematical Programming in Montreal, August 1979, Edmonds and Lovász conjectured that the results of this paper when combined with Khachian's algorithm [4] for linear programming problems yield a polynomial time and space matching algorithm. In [17] we show that the complexity of the resulting (dual) algorithm is $0(n^7)$.

Acknowledgement. The authors acknowledge some useful discussions with W. H. Cunningham that led to a shortening of the proof of Lemma 1.1.

References

[1] Balinski, M. L. (1972). Establishing the Matching Polytope. *J. Combinatorial Theory Ser. B.* **13** 1–13.

[2] Crowder, H. P. and Padberg, M. W. (1980). Solving Large-Scale Symmetric Travelling Salesman Problems to Optimality. *Management Sci.* **26** 495–509.

[3] Cunningham, W. H. and Marsh, A. B., III. (1978). A Primal Algorithm for Optimum Matching. *Mathematical Programming Study* **8** 50–72.

[4] Edmonds, J. (1965). Paths, Trees, and Flowers. *Canad. J. Math.* **17** 449–467.

[5] ———. (1965). Maximum Matching and a Polyhedron with 0, 1 Vertices. *J. Res. Nat. Bur. Standards Sect. B.* **69** 125–130.

[6] ———. (1967). An Introduction to Matching. Lecture Notes, The University of Michigan, Ann Arbor.

[7] Edmonds, J. (1971). Matroids and the Greedy Algorithm. *Math. Programming* **1** 127–136.
[8] ——— and Johnson, E. L. (1970). Matching: A Well Solved Class of Integer Linear Programs. In *Combinatorial Structure and Their Applications*, R. Guy, ed. Gordon and Breach, New York.
[9] ———, ——— and Lockhart, S. (1969). Blossom I, A Code for Matching. Unpublished report, IBM T. J. Watson Research Center, Yorktown Heights.
[10] Ford, L. R., Jr. and Fulkerson, D. R. (1962). *Flows in Networks.* Princeton University Press, Princeton.
[11] Gomory, R. E. and Hu, T. C. (1961). Multi-terminal Network Flows. *SIAM J. Appl. Math.* **9** 551–556.
[12] Hu, T. C. (1969). *Integer Programming and Network Flows.* Addison-Wesley, Reading.
[13] Johnson, E. L. (1965). Networks, Graphs and Integer Programming. Ph.D. thesis, Operations Research Center, The University of California, Berkeley.
[14] Khachian, L. G. (1979). A Polynomial Algorithm in Linear Programming. *Doklady Akad. Nauk USSR.* **244**; translated in *Soviet Math. Doklady* **20** 191–194.
[15] Lawler, E. L. (1976). *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston, New York.
[16] Padberg, M. W. and Hong, S. (1980). On the Symmetric Travelling Salesman Problem: A Computational Study. *Mathematical Programming Studies* **12** 199–215.
[17] ——— and Rao, M. R. (January 1980). The Russian Method and Integer Programming. GBA Working Paper, New York University, New York.
[18] Pulleyblank, W. R. (1973). Faces of Matching Polyhedra. Ph.D. thesis, The University of Waterloo.

GRADUATE SCHOOL OF BUSINESS ADMINISTRATION, NEW YORK UNIVERSITY, NEW YORK, NEW YORK 10006