Appa, Gautam and Kotnyek, Balázs

Optimization with binet matrices

Working paper

Optimization with binet matrices

Gautam Appa[*]
London School of Economics

Balázs Kotnyek[†]
I3S & Inria Sophia Antipolis

December 15, 2003

Abstract

This paper deals with linear and integer programming problems in which the constraint matrix is a binet matrix. Binet matrices are pivoted versions of the node-edge incidence matrices of bidirected graphs. It is shown that efficient methods are available to solve such optimization problems. Linear programs can be solved with the generalized network simplex method, while integer programs are converted to a matching problem. It is also proved that an integral binet matrix has strong Chvátal rank 1. An example of binet matrices, namely matrices with at most three non-zeros per row, is given.

Key words: half-integrality, generalized networks, matching, strong Chvátal rank

1 Introduction

If the constraint matrix of a linear program is a network matrix, then we can use the network simplex algorithm to find an optimal solution much more efficiently compared to the general-purpose simplex method. In this paper, we shall see that this advantage of network matrices can be extended to their bidirected generalization, the binet matrices.

Network matrices, a well-known class of totally unimodular matrices, are defined on directed graphs. This graphical structure behind network matrices made it possible to devise the network simplex method, a successful cross between the algebra of the simplex method and the combinatorics of the network flow algorithms. It has been reported in the literature (e.g., in [10]) that the network simplex method can be up to 200 times faster than the general-purpose linear programming codes. Network matrices and the network simplex method have become integral parts of the combinatorial optimization apparatus, and they are treated in every textbook in this area, such as, for example [1],[15] and [16].

Appa and Kotnyek generalized network matrices in [2, 3] by introducing binet matrices. A binet matrix is defined on a bidirected graph in a way similar to how a network matrix is defined on a directed graph. They also showed that a binet matrix is 2-regular, i.e., the inverse of any of it's non-singular submatrix is half-integral. This fact, together with a theorem proved in [3], implies that if the constraint matrix of a linear program is a binet matrix, then all basic optimal solutions are half-integral. However, the question of how to find an optimal solution in an efficient way was left open. We shall deal with it in this paper.

Let us designate a linear programming problem with a binet constraint matrix a binet LP and an integer programming problem with a binet constraint matrix a binet IP. We shall see that efficient algorithms exist that

[*]Corresponding author: Gautam Appa, LSE, OR Dept., Houghton Street, London WC2A 2AE, g.appa@lse.ac.uk

[†]The author's research was partially supported by the European FET project Cressco, RTN project Aracne, and the French CNRS AS Dynamo

can find the optimal solution to a binet LP or IP. Section 3 is about finding optimal solutions to a binet LP. The main discovery is that a binet LP can be seen as a generalized network flow problem, therefore the existing generalized network simplex algorithm can be applied to it. As to binet IP, treated in in Section 4, we show that it can be converted to a generalized matching problem. Moreover, we prove that the integer hull of the polyhedron in a binet IP problem can be achieved by rank-1 cuts. In other words, an integral binet matrix has strong Chvátal rank 1.

Numerous examples of binet matrices can be found in [2, 12]. These include two interesting non-network but totally unimodular matrices, all the known minimally non totally-unimodular matrices and generalised interval matrices. Here, in Section 5 we show that the constraint matrix of special integer programs with at most three non-zeros per row, shown in [11] to have many practical examples, is a binet matrix.

2 Preliminaries

In this section we give a concise description of binet matrices and list some relevant results about them. More details can be found in [2] and [12].

A bidirected graph can be defined with its node-edge incidence matrix. Given an $m \times n$ integral matrix $A = (a_{ij})$ satisfying

$$\sum_{i=1}^{m} |a_{ij}| \leq 2 \quad \text{for } j = 1, \ldots, n, \tag{1}$$

we can find a bidirected graph $G(A)$ with m nodes and n edges such that its node-edge incidence matrix is A under the follwoing conventions. Edges of $G(A)$ correspond to columns, and nodes to rows, of A. An edge is incident to a node if the corresponding cell of the matrix contains a non-zero. Columns with a single non-zero being ± 2 represent loops, i.e., edges whose end-nodes coincide; columns with a single ± 1 represent half-edges with only one end-node. Positive and negative entries in the matrix correspond to heads and tails of edges, respectively. That is, an edge of a bidirected graph can have at most two heads and at most two tails, with the condition that if the edge is a loop than both of its ends, which are incident to the same node, are of the same kind.

Let $A = [S, R]$ be a full row rank node-edge incidence matrix of a bidirected graph, such that R is a basis of A. The matrix $B = R^{-1}S$ is called a *binet matrix*. Binet matrices can be achieved by a series of pivoting on the incidence matrix, and they can contain elements $\{0, \pm ½, \pm 1, \pm 2\}$.

Several operations on a binet matrix preserve its binetness. For example, after adding unit rows or columns, repeating or deleting rows or columns, a binet matrix remains binet.

Binet matrices are 2-regular, i.e., the inverse of any nonsingular submatrix is half-integral. It was proved in [3] that 2-regularity of the constraint matrix implies half-integral optimal solutions to a linear program, i.e. solution vectors that contain integers and halves of integers. Specifically for binet matrices:

Theorem 1. *If B is a binet matrix and l, u, a, b are integer vectors of appropriate size, then the basic solutions of the optimization problem* $\max\{cx \mid l \leq x \leq u,\ a \leq Bx \leq b\}$ *are all half-integral.*

Is it worth pointing out that if B is not a binet matrix but a general non-integral and 2-regular matrix, then only a weaker version of Theorem 1, in which we deal with $\max\{cx \mid x \geq 0,\ Bx \leq b\}$, holds.

3 Binet LPs

Let B be a binet matrix. We shall deal with the following binet LP problem:

$$\max\{cx \mid l \leq x \leq u,\ a \leq Bx \leq b\} \tag{2}$$

The optimal solution of (2) can be achieved by general-purpose methods, like the simplex algorithm, or the strongly polynomial algorithm of Tardos [17]. This latter states that for rational linear programming problems in which the elements of the constraint matrix are bounded, there exists an algorithm to solve the problem that uses arithmetic operations whose number is a polynomial function of the dimension of the problem and which act on rationals of size polynomially bounded by the size of the input. More about this ingenious algorithm can be found in [16]. Since the elements of a binet matrix are between -2 and 2, Tardos' algorithm on problems with binet constraint matrix have a strongly polynomial worst-case running time. However, despite this attractive theoretical complexity result, the up-to-date implementations of the simplex algorithm usually outperform Tardos' strongly polynomial method on practical instances.

Alternatively, we can apply the following transformation to (2). Let binet matrix $B = R^{-1}S$ where $[S, R]$ is the node-edge incidence matrix of a bidirected graph. Let us introduce new variables $z = -Bx$ to get:

$$\max\{cx + \mathbf{0}z \mid l \leq x \leq u,\ -b \leq z \leq -a,\ Bx + z = \mathbf{0}\} \tag{3}$$

Multiplying the equality by R we get

$$\max\{cx + \mathbf{0}z \mid l \leq x \leq u,\ -b \leq z \leq -a,\ Sx + Rz = \mathbf{0}\} \tag{4}$$

The constraint matrix of the latter problem, which is equivalent to (2), is the node-edge incidence matrix of a bidirected graph, therefore it satisfies (1). We shall see now that problems with such constraint matrices can be solved by a special version of the simplex method that exploits the generalized network structure of the constraint matrix.

A *generalized network* is defined on a connected directed graph G. There is a real non-zero multiplier p_e associated with each edge $e = (i, j)$ of G. We assume that if a unit flow leaves the tail i of e, then p_e units arrive at j. G can also contain loops, i.e., edges whose tail and head nodes coincide. We assume that the multiplier of a loop cannot be $+1$, as it would mean that the same flow leaves and enters the node on such a loop, making the loop redundant. Trivially, if all the multipliers equal 1, then we have the well-known pure network. Generalized networks are discussed in more detail in [1, 14].

A way of describing a generalized network is with its node-edge incidence matrix. The column of the incidence matrix that corresponds to a non-loop edge $e = (i, j)$ has -1 in row i and p_e in row j, zeros elsewhere. If e is a loop at node i, then its column has only one non-zero, $(p_e - 1)$ in row i. With simple operations, every matrix with at most two non-zeros per column can be converted to this form. In what follows, we show how this can be done for the node-edge incidence matrices of bidirected graphs.

According to (1), the node-edge incidence matrix of a bidirected graph can have columns of the following forms:

(a) columns with one non-zero, being $+1$ or ± 2,
(b) columns with one -1 and another non-zero, being ± 1,
(c) columns with one non-zero, being -1,
(d) columns with two non-zeros, both being 1.

Columns of types (a) and (b) can be columns of the node-edge incidence matrix of a generalized network: columns of type (a) are loops, columns of type (b) are ordinary non-loop edges with $p_e = \pm 1$. Columns with a single non-zero -1 cause a problem, because representing them with $(p_e - 1)$ would require $p_e = 0$. But this problem can be easily overcome by multiplying such columns by 2, obtaining a column of type (a). The multiplication of a column by 2 is equivalent to dividing the corresponding variable by 2. Columns of the last type are also problematic, as an edge corresponding to such a column cannot be a generalized network edge as defined in the current literature, as this edge would have no tail. We provide now a simple transformation to deal with columns of type (d).

Let us suppose that we have a binet LP problem

$$\begin{aligned} &\min c_1x_1 + c_2x_2 \\ \text{subject to}\quad &A_1x_1 + A_2x_2 = b \\ &0 \leq x_1 \leq u_1 \\ &0 \leq x_2 \leq u_2 \end{aligned} \tag{5}$$

where $A = [A_1, A_2]$ is the node-edge incidence matrix of a bidirected graph with A_1 containing the columns of type (d). Then with the new variable $x_1' = -x_1 + u_1$ we get the equivalent binet LP problem

$$\begin{aligned} &\min -c_1x_1' + c_1u_1 + c_2x_2 \\ \text{subject to}\quad &-A_1x_1' + A_2x_2 = b - A_1u_1 \\ &0 \leq x_1' \leq u_1 \\ &0 \leq x_2 \leq u_2 \end{aligned} \tag{6}$$

in which all columns are of types (a), (b) or (c), so the constraint matrix can be considered the node-edge incidence matrix of a generalized network.

The idea behind the network simplex method is that the main steps of the simplex method (such as calculating the primal and dual solutions corresponding to a basis, or changing the basis) can be executed on the network associated with the constraint matrix. This idea can be followed in generalized networks too, leading to the *generalized network simplex method*. The textbook references given above, [1, 14], contain a detailed description of the generalized network simplex method. They also give references to papers that deal with the problem of finite termination of the method. It should be noted that the generalized network simplex method is not polynomial in the worst case, but for most of the practical problems it is much more efficient than the simplex method, or the strongly polynomial method mentioned above (see the reference notes in [1]).

As argued in [12], the generalized network simplex method described in the references can be easily adapted to the case where the network contains edges with two heads and no tails (i.e., edges corresponding to columns of type (d)). Therefore the generalised network simplex method works on bidirected graphs too, making the column transformation described above unnecessary. Either way, with column transformation or by modifying the algorithm, the generalized network simplex method can be seen as the bidirected version of the network simplex algorithm, able to solve the binet LP problem much more efficiently than a general purpose simplex algorithm.

4 Binet IPs

Now we turn to the integer case, that is, we are to solve the following binet IP problem.

$$\max\{cx \mid l \leq x \leq u,\ a \leq Bx \leq b,\ x \text{ integral}\} \tag{7}$$

in which B is an integral binet matrix and l, u, a, b are integral vectors. With the transformation described in (3) and (4), noticing that the integrality of B implies the integrality of z, we have the following equivalent version of (7):

$$\max\{cx + \mathbf{0}z \mid l \leq x \leq u,\ -b \leq z \leq -a,\ Sx + Rz = \mathbf{0},\ x, z \text{ integral}\} \tag{8}$$

By translations, i.e., substituting $x - l$ and $z + b$ for x and z, respectively, (8) can be brought to the form of

$$\max\{wy \mid 0 \leq y \leq h,\ Ay = d,\ y \text{ integral}\} \tag{9}$$

This problem can be viewed as a *bidirected network flow problem*, the aim of which is to find the maximum weight integer flow in a bidirected network that satisfies the net supply (or demand) and the capacity conditions. Matrix A is the node-edge incidence matrix of the bidirected graph, y represents the flow on the edges, w is the weight vector, h is the capacity vector and d represents the net supply (or demand) at the nodes. The bidirected network flow problem was introduced by Edmonds [6]. He also showed that it is equivalent to a general matching problem (see also [13]).

Edmonds and Johnson [7] showed that there exists a polynomial algorithm to solve general matching problems, even on bidirected graphs. Thus, as the integer binet optimization problem is essentially a bidirected network flow problem and the bidirected network flow problem is in effect a general matching problem, we have the following result.

Theorem 2. *There is a strongly polynomial algorithm to solve the binet IP problem.*

Edmonds and Johnson also gave polyhedral results in [7, 8]. They showed that the node-edge incidence matrix of a bidirected graph has strong Chvátal rank 1, that is, the integer hull of a general polyhedron defined by the incidence matrix of a bidirected graph and any integer right hand side vector can be achieved by rank-1 cuts. (That is why matrices with strong Chvátal rank 1 are sometimes said to have the Edmonds-Johnson property.) We extend this result to integral binet matrices.

Formally, let a rank-1 *Chvátal-Gomory (CG) cut* of polyhedron Q be defined as $cx \leq \lfloor \delta \rfloor$ where c is an integral vector and δ is a scalar such that $cx \leq \delta$ is valid for all $x \in Q$. The intersection of Q with the half-spaces induced by all the possible rank-1 CG-cuts is its *rank-1 closure*, denoted Q_1. An integral $m \times n$ matrix B has *strong Chvátal rank* 1, if $Q_1 = Q_I = conv(Q \cap \mathbb{Z}^n)$ for polyhedron $Q = \{x \mid l \leq x \leq u,\ a \leq Bx \leq b\}$ and all integral vectors l, u, a, b.

Theorem 3. *If B is an integral binet matrix, then it has strong Chvátal rank 1.*

Proof: Let $B = R^{-1}S$ and $A = [S, R]$ is the node-edge incidence matrix of the bidirected graph representing B. Let P be the polyhedron defined by the constraints in (7) and $\bar{P}$ is the polyhedron defined by the constraints in (8). Obviously, $x \in P$ if and only if $\begin{pmatrix} x \\ -Bx \end{pmatrix} \in \bar{P}$.

Let y not be in P_I. We show that then y can be cut off from P by a rank-1 CG cut, so that $P_1 \subseteq P_I$, which suffices to prove the theorem.

It is easy to show that $\begin{pmatrix} y \\ -By \end{pmatrix} \notin \bar{P}_I$. Because A has strong Chvátal rank 1, there exists an integral vector $\bar{c} = (c_1, c_2)$ and scalar δ such that $(c_1, c_2)\begin{pmatrix} x \\ z \end{pmatrix} \leq \delta$ is valid for all $\begin{pmatrix} x \\ z \end{pmatrix} \in \bar{P}$ and $(c_1, c_2)\begin{pmatrix} y \\ -By \end{pmatrix} > \lfloor \delta \rfloor$. Hence for the integral vector $c = c_1 - c_2 B$, inequality $cx \leq \delta$ is valid for all $x \in P$ and $cy > \lfloor \delta \rfloor$. $\square$

Note that Theorem 3 cannot be extended to rational binet matrices, as the following example shows.

$$B = R^{-1}S = \begin{bmatrix} \frac{1}{2} & 1 & 0 \\ 0 & 1 & 1 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} \text{ with } R = \begin{bmatrix} 1 & 0 & -1 \\ 1 & -1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, S = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

The non-integral binet matrix B does not have strong Chvátal rank 1, because the non-zero integral solutions of the polyhedron $P = \{x \mid 0 \le x \le 1,\ 0 \le Bx \le 1\}$ are $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$, so $x_1 + x_2 + x_3 \le 1$ is a facet of P_I. But $(1, ½, ½) \in P$, so $\delta = 2$ is the smallest value for which $x_1 + x_2 + x_3 \le \delta$ is valid for P.

Note that this counterexample shows that the strong Chvátal rank of binet matrices is not simply a consequence of the fact that binet matrices are pivoted versions of node-edge incidence matrices of bidirected graphs. If this was the case, i.e., pivoting would preserve the strong Chvátal rank, then any rational binet matrix should also have strong Chvátal rank 1.

In [9], Gerards and Schrijver gave a characterization of integral matrices whose transpose satisfies (1) and have strong Chvátal rank 1. In other words, they dealt with the *edge-node* incidence matrices of bidirected graphs. The key graph and matrix in their characterization is K_4, the undirected complete graph on four nodes, and its edge-node incidence matrix. They proved that the edge-node incidence matrix of a bidirected graph has strong Chvátal rank 1, if and only if the graph cannot be transformed to K_4 by a series of graph operations. A consequence of this result and Theorem 3 is that if a graph can be so transformed to K_4, then its edge-node incidence matrix is not binet, though its transpose, the node-edge incidence matrix is clearly binet.

In [3] the authors proved that if A is an integral 2-regular matrix of size $m \times n$, then for polyhedron $Q = \{x \mid Ax \le b, x \ge 0\}$ with integral b, the rank-1 closure Q_1 can be achieved by only half-integral cuts, i.e., valid inequalities of the form $\lambda A x \le \lfloor \lambda b \rfloor$ where $\lambda \in \{0, ½\}^m$ and λA is integral. In a compact form, this result states that $Q_1 = Q_{½}$ for integral 2-regular matrices and any integral right hand side vector, if we define $Q_{½}$ as the $\{0, ½\}$-closure of Q, i.e., the intersection of Q with the half-spaces induced by all the possible half-integral cuts. As binet matrices are 2-regular, we immediately get the following corollary of Theorem 3.

Corollary 4. *If B is an integral binet matrix and b is an integral vector, then the integer hull of $Q = \{x \mid Bx \le b, x \ge 0\}$ can be achieved by half-integral cuts, i.e., $Q_I = Q_{½}$.*

This result has an interesting consequence in separation. The $\{0, ½\}$-*separation problem*, as defined in [5], is the following:

> Given $x \in Q = \{x \mid Ax \le b, x \ge 0\}$, decide if x is in $Q_{½}$ or not, and if it is not, find a half-integral cut that separates it, i.e., a $\lambda \in \{0, ½\}^m$ such that $\lambda A \in \mathbb{Z}^n$ and $\lambda A x > \lfloor \lambda b \rfloor$.

It is well known (see e.g., [16]), that the separation problem is polynomially equivalent to the optimization problem. In the special case of $\{0, ½\}$-separation, it means that if we can optimize linear functions over $Q_{½}$ in polynomial time, then we can decide the separation question in polynomial time. As we showed above, the integer optimization (i.e., optimizing linear functions over Q_I) with integral binet constraint matrices is polynomial, since it is equivalent to a matching problem, so we have the following consequence of Corollary 4:

Corollary 5. *If A is an integral binet matrix, then the $\{0, ½\}$-separation problem can be solved in polynomial time.*

If A or its transpose is a network matrix, then the $\{0, ½\}$-separation is trivial, as $Q_{½} = Q$. This is because for totally unimodular matrices $Q_I = Q$, and for any polyhedron $Q_I \subseteq Q_{½} \subseteq Q$. Corollary 5 extends this result to integral binet matrices.

5 Matrices with at most three non-zeros per row

Hochbaum [11] examines integer programs in which each constraint involves up to three variables. That is, a constraint that is not an upper bound on a variable can be of the forms:

$$a_i x_i + a_j x_j \le b \quad \text{or} \quad a_i x_i + a_j x_j + z_{ij} \le b$$

where a_i, a_j are rationals without any restriction on their signs. A further assumption is that variable z_{ij} appears in only one constraint. Following [11], we will call such a problem an *IP2 problem*. The generic matrix format of IP2 problems is the following:

$$\begin{aligned} &\max cx + dz \\ &\begin{bmatrix} A_1 & I \\ A_2 & 0 \end{bmatrix} \begin{pmatrix} x \\ z \end{pmatrix} \leq b \\ &l \leq x \leq u \\ &l_z \leq z \leq u_z \end{aligned} \tag{IP2}$$

where matrices A_1 and A_2 contain at most two non-zeros in each row. It is also permitted to add further identity matrices while maintaining the results. That is, the constraint matrix can be of the form:

$$\begin{bmatrix} A_1 & I & \cdots & I \\ A_2 & 0 & \cdots & 0 \end{bmatrix}$$

A special IP2 problem, called *binarized*, is when the elements of the constraint matrix are all $0, \pm 1$. Hochbaum [11] concludes that a binarized IP2 problem always has half-integral basic solutions, and builds 2-approximation methods based on this fact. In other words, she proves that the constraint matrix of (IP2) is 2-regular. Here we show that the constraint matrix of a binarized IP2 problem is the transpose of a binet matrix, therefore its 2-regularity immediately follows.

Theorem 6. *The transpose of the constraint matrix of a binarized IP2 problem is binet.*

Proof: Matrix $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ has at most two non-zeros in each row, and these non-zeros are ± 1, so A^T satisfies (1). So does $[A^T, I]$ too, where I is a unit matrix of appropriate size. Therefore $[A^T, I]$ is the node-edge incidence matrix of a bidirected graph, and $A^T = I^{-1}A^T$ is a binet matrix. As mentioned in Section 2, adding unit rows to a binet matrix maintains its binetness, so adding unit columns to A does not change the fact that it is the transpose of a binet matrix. □

Hochbaum also gives combinatorial applications for the binarized IP2 problem, including the feasible cut problem, the complement of the maximum clique problem, the generalized independent set problem and the generalized vertex cover problem. These applications are thus examples of combinatorial problems with binet matrices. For example, we describe the generalized independent set problem. Details of this and the other problems can be found in [11].

The *generalized independent set problem* is the generalization of the well-known independent set problem. In the latter, the aim is to find a maximum weight node set in a graph $G(V, E)$, such that there is no edge between the selected nodes. In the generalized version, we permit edges, but at a penalty. The IP2 formulation of the generalized independent set problem is:

$$\begin{aligned} \max \quad & \sum_{i \in V} w_i x_i - \sum_{(i,j) \in E} c_{ij} z_{ij} \\ \text{subject to} \quad & x_i + x_j - z_{ij} \leq 1 \quad \forall (i,j) \in E \\ & x_i, z_{ij} \in \{0, 1\} \quad \forall i, j \end{aligned}$$

Variables x_i ($i \in V$) represent nodes; $x_i = 1$ if and only if node i is selected. For each edge $e = (i, j)$, we have a variable z_{ij}, and the constraints of the model guarantee that if both end-points of e are selected, then $z_{ij} = 1$. The weight of node i is w_i, the penalty on edge (i, j) is c_{ij}.

A real-life application of the generalized independent set problem concerns the location of postal services [4]. We are given a set of potential location points and the utility value (weight) associated with each point. If two points are too close to each other, then they compete for the same costumers, so their utility value is decreased. This is the penalty on the edge connecting these points. The goal is to find a set of locations that maximizes the utility value.

References

[1] R.K. Ahuja, Th.K. Magnanti, and J.B. Orlin. *Network Flows*. Prentice Hall, New Jersey, 1993.

[2] G. Appa and B. Kotnyek. A bidirected generalization of network matrices. *submitted to Networks*, 2003.

[3] G. Appa and B. Kotnyek. Rational and integral k-regular matrices. *Discrete Mathematics*, 2003.

[4] M. Ball. Locating competitive new facilities in the presence of existing facilities. In *Proceedings of the 5th United States Postal Service Advanced Technology Conference*, pages 1169–1177, 1992.

[5] A. Caprara and M. Fischetti. $\{0, \frac{1}{2}\}$-Chvátal-Gomory cuts. *Mathematical Programming*, 74:221–235, 1996.

[6] J. Edmonds. An introduction to matching. Mimeographed notes, Engineering Summer Conference, The University of Michigan, 1967.

[7] J. Edmonds and E.L. Johnson. Matching: a well-solved class of integer linear programs. In R.K. Guy et al., editors, *Combinatorial Structures and Their Applications*. Gordon and Breach, New York, 1970.

[8] J. Edmonds and E.L. Johnson. Matching: Euler tours and the Chinese postman. *Mathematical Programming*, 5:88–124, 1973.

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

[10] F. Glover, D. Karney, and D. Klingman. Implementation and computational comparisons of primal, dual and primal-dual computer codes for minimum cost network flow problem. *Networks*, 4:191–212, 1974.

[11] B.S. Hochbaum. Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. *European Journal of Operational Research*, 140(2):291–321, 2002.

[12] B. Kotnyek. *A generalization of totally unimodular and network matrices*. PhD thesis, London School of Economics, 2002.

[13] E.L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart & Winston, New York, 1976.

[14] K.G. Murty. *Network Programming*. Prentice Hall, New Jersey, 1992.

[15] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1988.

[16] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.

[17] É. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34:250–256, 1986.