Instant recognition of half integrality and 2-approximations

Dorit S. Hochbaum

Department of Industrial Engineering and Operations Research, and Walter A. Haas School of Business, University of California, Berkeley

Abstract. We define a class of integer programs with constraints that involve up to three variables each. A generic constraint in such integer program is of the form $ax + by \leq z + c$, where the variable z appears only in that constraint. For such binary integer programs it is possible to derive half integral superoptimal solutions in polynomial time. The scheme is also applicable with few modifications to nonbinary integer problems. For some of these problems it is possible to round the half integral solution to a 2-approximate solution. This extends the class of integer programs with at most two variables per constraint that were analyzed in [HMNT93]. The approximation algorithms here provide an improvement in running time and range of applicability compared to existing 2-approximations. Furthermore, we conclude that problems in the framework are MAX SNP-hard and at least as hard to approximate as vertex cover.

Problems that are amenable to the analysis provided here are easily recognized. The analysis itself is entirely technical and involves manipulating the constraints and transforming them to a totally unimodular system while losing no more than a factor of 2 in the integrality.

1 Introduction

We demonstrate here for a given class of integer programming problems a unified technique for deriving constant approximations and superoptimal solutions in half integers. The class is characterized by formulations involving up to three variables per constraint one of which appears in a single constraint. Such problem formulations have a number of interesting properties that permit the derivation of lower or upper bounds that are of better quality than the respective linear programming relaxations, and that can be turned in some cases into feasible solutions that are 2-approximate.

The integer programs described here have a particularly structured set of constraints, with a generic constraint of the form $ax + by \leq z + c$. Note that any linear optimization problem, integer or continuous, can be written with at most three variables per inequality. The generic constraints here have one of

* Research supported in part by NSF award No. DMI-9713482, and by SUN Microsystems.

the variables appearing only once in a constraint. We call such structure of the set of constraints the 2var structure. The paper describes an analysis of such problems based on manipulating the constraints and transforming them to a totally unimodular system. The outcome of this process is a polynomial algorithm that delivers a superoptimal solution to all integer programs with 2var constraints that has the variables on the left hand side as integer multiple of $\frac{1}{2}$, and the variables on the right hand side, z as an integer multiple of $\frac{1}{2}\frac{1}{U}$ where U is the number of integer values in the range of the variables x and y. In the case of binary problems there is an exception with the value $U = 2$ with z being an integer multiple of $\frac{1}{2}$ (although there are two integer values for each variable). Being superoptimal means that the solution's objective value can only be better (lower) than the optimum (for minimization problems).

When it is possible to round the $\frac{1}{2}$ integral solution to a feasible solution then a 2-approximate solution is obtained.

Applications and 2-approximations based on the technique. The class of problems addressed by the technique described expands substantially the class treated in [HMNT93]. There we demonstrated that $\frac{1}{2}$ integral solutions and 2-approximations are *always* obtained in polynomial time for minimization (with nonnegative coefficients) integer programs with at most *two* variables per inequality. For these problems a feasible rounding always exists provided that the problems are feasible. The technique of [HMNT93] was shown applicable to the vertex cover problem and the problem of satisfying a 2SAT formula with the least weight of true variables. Here and in other papers we demonstrate 2-approximations for additional problems:

- **minimum satisfiability**. In the problem of *minimum satisfiability* or MIN-SAT, we are given a CNF satisfiability formula. The aim is to find an assignment satisfying the smallest number of clauses, or the smallest weight collection of clauses. The minimum satisfiability problem was introduced by Kohli et. al. [KKM94] and was further studied by Marathe and Ravi [MR96]. Marathe and Ravi discovered a 2-approximation algorithm to the problem, that can be viewed as a special case of our general algorithm for problems with two variables per inequality.
- **Scheduling with precedence constraints**, [CH97].
- **Biclique problems**. Minimum weight node deletion to obtain a complete bipartite subgraph – biclique [Hoc97]. With the use of the technique described here we identified the polynomiality of the problem on general graphs when each side of the biclique is not required to form an independent set. When this requirement is introduced the problem is NP-hard and a 2-approximation algorithm is given. Another variant of the biclique problem is the minimum weight edge deletion. This problem is NP-hard even on a bipartite graphs. All variants of this problem considered in [Hoc97] are NP-hard and 2-approximable.
- **The complement of the edge maximum clique problem**. This problem is to minimize the weight of the edges deleted so the remaining subgraph is a clique, [Hoc97].

- **Generalized satisfiability problems**, [HP97].
- **The generalized vertex cover problem**, [Hoc97a].
- **The t-vertex cover problem**. [Hoc98].
- **The feasible cut problem**. The 2-approximation for this problem is illustrated here.

The last three items on the list are problems that have three variables per inequality. The technique delivers $\frac{1}{2}$ integral solutions as well as polynomial time 2- approximation algorithms for these problems. All the approximations listed are obtained using a minimum cut algorithm, and thus in strongly polynomial time.

The problems that fall in the class described here and are NP-hard are then provably at least as hard to approximate as the vertex cover problem (see [HMNT93] and [Hoc96]). Therefore an approximation factor better than 2 is only possible provided that there is such approximation for the vertex cover.

1.1 The Structure of 2var-Constraints

2var constraints contain two types of integer variables, a vector $\mathbf{x} \in \mathbb{N}^{\mathbf{n}}$ and $\mathbf{z} \in \mathbb{N}^{\mathbf{m_2}}$. We refer to them here as x-variables and z-variables respectively. A 2var constraint has at most two x-variables appearing in it. The z-variables can appear each in just one constraint.

We refer to integer problems of optimizing over 2var constraints as IP2. A formulation of a typical IP2 is,

$$
\text{(IP2)} \qquad
\begin{array}{ll}
\text{Min} & \sum_{j=1}^{n} w_j x_j + \sum e_i z_i \\
\text{subject to} & a_i x_{j_i} + b_i x_{k_i} \geq c_i + d_i z_i \quad \text{for } i = 1, \dots, m \\
& \ell_j \leq x_j \leq \; u_j \quad j = 1, \dots, n \\
& z_i \;\; \text{integer} \qquad i = 1, \dots, m_2 \\
& x_j \;\; \text{integer} \qquad j = 1, \dots, n.
\end{array}
$$

A parameter of importance in the analysis is the number of values in the range of the x-variables, $U = max_{j=1,\dots n}(u_j - \ell_j + 1)$. We assume throughout that U is polynomially bounded thus permitting a reference to running time that depends polynomially on U as polynomial running time.

An important property of 2var inequalities that affect the complexity of the IP2 is monotonicity:

Definition 1. *An inequality $ax - by \leq c + dz$ is* **monotone** *if $a, b \geq 0$ and $d = 1$.*

The IP2 problem was studied for $D = \max |d_i| = 0$ in [HN94] and in [HMNT93]. For such problems there are only x-variables and at most two of them per inequality. Hochbaum and Naor [HN94] devised a polynomial time algorithm to solve the problem in integers over *monotone* inequalities. For (nonmonotone) inequalities with at most two variables per inequality, $D = 0$, Hochbaum, Megiddo, Naor and Tamir described a polynomial time 2-approximation algorithm. These results are extended here for $D \geq 1$.

An important special case of IP2 where the value of U affects neither the complexity nor the approximability is of binarized IP2:

Definition 2. *An IP2 problem is said to be* **binarized** *if all coefficients of the variables are in* $\{-1, 0, 1\}$*. Or, if* $B = \max_i\{|a_i|, |b_i|, |d_i||\} = 1$*.*

Note that a binarized system is not necessarily defined on binary variables. The value of U may be arbitrarily large for a binarized IP2 without affecting the polynomiality of the results as proved in our main theorem.

1.2 The Main Theorem

For IP2 over *monotone* constraints that is binarized we describe a polynomial time algorithm solving the problem in integers. The polynomial algorithm for the monotone IP2 is used as a building block in the derivation of the superoptimal solution and approximations of nonmonotone problems.

For nonmonotone constraints a polynomial time procedure derives a superoptimal half integral solution for the x-variables in which the z-variables are an integer multiple of $\frac{1}{2DU}$. If that solution has a rounding to a feasible integer solution, and the objective function coefficients are nonnegative then that solution is a 2-approximation, or for arbitrary D and U it is a $2DU$-approximation.

For an integer program over 2var constraints, the running time required for finding a superoptimal half integral solution can be expressed in terms of the time required to solve a linear programming over a totally unimodular constraint matrix, or in terms of minimum cut complexity. In the complexity expressions we let $T(n, m)$ be the time required to solve a minimum cut problem on a graph with m arcs and n nodes. $T(n, m)$ may be assumed equal to $O(mn \log(n^2/m))$, [GT88]. For binarized system the running time depends on the complexity of solving a minimum cost network flow algorithm $T_1(n, m)$. We set $T_1(n, m) = O(m \log n(m + n \log n))$, the complexity of Orlin's algorithm [Orl93].

Let $T = T(2\sum_{j=1}^{n}[u_j - \ell_j], 2m_1(U - 1) + 2m_2U^2)$ and $T_1 = T_1(2\sum_{j=1}^{n}[u_j - \ell_j], 2m_1U + 2m_2(U + 1)^2)$. The main theorem summarizing our results is,

Theorem 3. *Given an* IP2 *on* $m = m_1 + m_2$ *constraints,* $\mathbf{x} \in \mathbf{Z^n}$ *and* $U = max_{j=1,\ldots n}(u_j - \ell_j + 1)$*.*

1. *A monotone IP2 with* $D \leq 1$ *is solvable optimally in integers for*
   - $U = 2$*, in time* $T(n, m)$*.*
   - *Binarized IP2 in time* $T_1(n, m)$*.*
   - *Otherwise, in time* T*.*
2. *For nonmonotone IP2 with* $D \leq 1$*, a superoptimal fractional solution is obtainable in polynomial time:*
   - *For* $U = 2$ *a half integral superoptimal solution is obtained in time* $T(2n, 2m)$*.*

– *For binarized IP2, a half integral superoptimal solution is obtained in time $T_1(2n, 2m)$.*
– *Otherwise, a superoptimal solution that is integer multiple of $\frac{1}{2}$ for the x-variables, and integer multiple of $\frac{1}{2U}$ for the z-variables is obtained in time T.*

3. *Given an* IP2 *and an objective function* $\min \mathbf{wx} + \mathbf{cz}$ *with* $\mathbf{w}, \mathbf{c} \geq \mathbf{0}$.
   – *For $D = 0$, if there exists a feasible solution then there exists a feasible rounding of the $\frac{1}{2}$ integer solution, which is a 2-approximate solution obtainable in time T, [HMNT93].*
   – *For binarized IP2, if there exists a feasible rounding of the fractional solution, then any feasible rounding is a 2-approximate solution obtainable in time T_1.*
   – *For $D > 0$, if there exists a feasible rounding of the fractional solution, then any feasible rounding is a $2DU$-approximate solution obtainable in time T. If $U = 2$ the solution is $2D$-approximate.*

2 The Algorithm

The algorithm solves the monotone IP2 problem in polynomial time using minimum cut or maximum flow algorithm. This is done by transforming the monotone constraints to an equivalent system that is totally modular but contain a larger number of constraints by a factor of U.

For NP-hard instances of IP2, those containing nonmonotone constraints, we employ a *monotonizing* procedure. This procedure is a transformation of the 2var constraints to another set of constraints with totally unimodular matrix coefficients. The transformed problem can then be solved in integers. The transformation back to the original polytope maps integers to half integers for the x-variables. Those can be rounded, under certain conditions, to a feasible solution within a factor of 2 of the optimum.

Algorithm IP2 described in Figure 1 works in two phases: Phase applies a process of "monotonizing" (step 1) the inequalities and "binarizing" (step 2) the variables. The second phase recovers the values of the fractional solution that is half integral for the x-variables and an integer multiple of $\frac{1}{2DU}$ for the z-variables.

2.1 Monotonizing

Consider first a generic nonmonotone inequality $ax + by \leq c + dz$. It can be assumed that z is scaled so that $d > 0$ and its objective function coefficient is positive. If the inequality is reversed, $ax + by \geq c + dz$, z is simply set to its lower bound. Replace each variable x by two variables, x^+ and x^-, and each term dz by z' and z''. The nonmonotone inequality is then replaced by two monotone inequalities:

$$ax^+ - by^- \leq c + z', \quad -ax^- + by^+ \leq c + z''.$$

The upper and lower bounds constraints $\ell_j \leq x_j \leq u_j$ are transformed to

Algorithm $\mathsf{IP2}$ ($\min\{\mathbf{ay} : \mathbf{By} \leq \mathbf{c}\}$)

1. **Monotonize** using the map $f : \mathbf{y} \rightarrow \mathbf{y}^+, \mathbf{y}^-$, $B'\mathbf{y} \leq \mathbf{c}'$.
2. **Binarize** , $B''\mathbf{y} \leq \mathbf{0}$
3. **Solve** $\min\{\mathbf{a}'\mathbf{y} : \mathbf{B}''\mathbf{y} \leq \mathbf{0}\}$ in integers. Let optimal solution be $\hat{\mathbf{y}}^+, \hat{\mathbf{y}}^-$.
4. Recover fractional solution $\hat{\mathbf{y}}$ to $\{B\mathbf{y} \leq \mathbf{c}\}$ by applying $f^{-1}(\hat{\mathbf{y}}^+, \hat{\mathbf{y}}^-)$.
5. **Round**. If a feasible rounding exists, round $\hat{\mathbf{y}}$ to $\mathbf{y}^*$.

Fig. 1. Schematic description of the algorithm $\mathsf{IP2}$

$$\ell_j \leq x_j^+ \leq u_j \qquad -u_j \leq x_j^- \leq -\ell_j \ .$$

In the objective function, the variable x is substituted by $\frac{1}{2}(x^+ - x^-)$ and z is substituted by $\frac{1}{2}(z' + z'')$.

Monotone inequalities remain so by replacing the variables x and y in one inequality by x^+ and y^+, and in the second, by x^- and y^-, respectively. The variable z is duplicated:

$$\begin{aligned} ax^+ - by^+ &\leq c + z' \\ ax^- - by^- &\leq c + z'' \ . \end{aligned}$$

It is easy to see that if $x^+, x^-, y^+, u^- z', z''$ solve the transformed system, then $x = \frac{1}{2}(x^+ - x^-), y = \frac{1}{2}(y^+ - y^-), z = \frac{1}{2d}(z' + z'')$ solve the original system.

2.2 Binarizing

The transformed monotonized and binarized system has the property that every extreme point solution is integer. The process of conversion of the system of inequalities to a system that has all coefficients in $\{0, -1, 1\}$ is referred to as *binarizing*. This process can be applied to a $\mathsf{2var}$ system of inequalities whether or not it is monotonized. For simplicity we describe the binarizing of monotonized inequalities.

"Binarizing" is a process transforming the monotonized system to inequalities of the form

$$x_i - x_j \leq 0 \text{ or } \ x_i - x_j \leq z_{ij} \ .$$

We start by replacing each x-variable x_i, $\ell \leq x_i \leq u$, by $u - \ell$ binary variables $x_i^{(\ell+1)}, \ldots, x_i^{(u)}$ so that $x_i = \ell + x_i^{(\ell+1)} + x_i^{(\ell+2)} + \ldots + x_i^{(u)}$, and so that $x_i \geq k x_i^{(k)}$. The values of $x_i^{(\ell+1)}, \ldots, x_i^{(u)}$ form a consecutive sequence of 1s followed by a consecutive sequence of 0s. Each of these sequences could possibly be empty. This structure is enforced by including for each variable x_i the inequalities,

$$x_i^{(k)} \geq x_i^{(k+1)} \qquad \text{for } k = \ell, \ldots, u-1, \tag{1}$$

where $x_i^{(\ell)} = 1$.

For a linear objective function with x_i's coefficient w, the term wx_i is replaced by,

$$w\ell + wx_i^{(\ell+1)} + \ldots + wx_i^{(u)}.$$

Binarizing $ax_i - bx_j \leq c$: Consider the monotone inequality $ax_i - bx_j \leq c$. This inequality enforces for each value $p_i \in [\ell_i, u_i]$ the implication: if $x_i \geq p_i$, x_j must satisfy for p_j a function of p_i,

$$x_j \geq \lceil \frac{ap_i - c}{b} \rceil \equiv p_j. \tag{2}$$

In other words, the implications

$$x_i^{(p_i)} = 1 \implies x_j^{(p_j)} = 1 \text{ for all } p_i \in [\ell_i, u_i]$$

are equivalent to the inequality $ax_i - bx_j \leq c$. If $p_j > u_j$ then the upper bound on x_i is updated by setting, $x_i \leq p_i - 1$ and $u_i \leftarrow p_i - 1$. To satisfy the set of implications it suffices to include the inequalities, $x_j^{(p_j)} \geq x_i^{(p_i)}$. We append the set of inequalities (1) with up to $\min\{u_i, u_j\}$ inequalities, $\forall p_i \in \{\ell_i, \ldots, u_i\}$ such that $p_j \in \{\ell_j, \ldots, u_j\}$,

$$x_j^{(p_j)} \geq x_i^{(p_i)}. \tag{3}$$

The set of inequalities (3) is equivalent to the inequality $ax_i - bx_j \geq c$.

Binarizing $ax_i - bx_j \leq dz + c$: Consider an inequality with three variables, $ax_i - bx_j \leq dz' + c$. First we substitute dz' by another variable z, thus deriving the inequality $ax_i - bx_j \leq z + c$. We assume that z's coefficient in the objective function is positive, else we can fix z at its upper bound, u_z, in an optimal solution.

Since the procedure is somewhat involved, we describe it here only for the case when z is binary. The general case is described in [Hoc97a].

z binary: Let p_j be a function of p_i as in (2). If $x_i^{(p_i)} = 1$ and

$$p_j - 1 = \lceil \frac{ap_i - c - 1}{b} \rceil < \lceil \frac{ap_i - c}{b} \rceil = p_j,$$

then, if $z = 1$, $x_j^{(p_j)}$ can be 0 (that is $x_j \leq p_j - 1$ rather than $x_j \geq p_j$). If the strict inequality is not satisfied then $z = 0$ in any optimal solution. The condition is enforced by setting $z = 1$ whenever $x_i^{(p_i)} = 1$ and $x_j^{(p_j)} = 0$, or equivalently, by satisfying the inequality $x_i^{(p_i)} - x_j^{(p_j)} \leq z$. Since in the case considered z cannot be great than 1, x_j must be at least $p_j - 1$ and we add the inequality $x_i^{(p_i)} \leq x_j^{(p_j-1)}$.

Now inequality $x_i^{(p_i)} - x_j^{(p_j)} \leq z$ with z in the right hand side may appear a number of times. There could be as many as U inequalities with z in the right hand side. Such set of inequalities is not totally unimodular. To convert it into a totally unimodular system we treat each occurrence of z as a separate variable,

$z_{(p_i)}$ and set $z = \sum_{\ell_i \le p_i \le u_i} z_{(p_i)}$. If there are several inequalities for z with the same p_j value then all but the one with the lowest p_i value are redundant. This argument leads to the conclusion that the number of inequalities can be no larger than U. Consequently, each occurrence of $z_{(p_i)}$ is assigned $\frac{1}{U}$ of the cost of z in the objective function. This guarantees that the objective function is a lower bound on the integer optimum.

2.3 The Network

Here we show that the totally unimodular system of inequality constraints is solvable by a minimum cut algorithm.

Theorem 4. *The integer problem on a monotone binarized system of inequalities is solvable by minimum s,t-cut algorithm on an appropriately defined network.*

The monotonized problem can be solved in integers using linear programming. Instead, we create a network where the source set of a minimum cut corresponds to an optimal solution to the monotonized binarized system.

Each binary variable $x_i^{(k)}$ has an associated node in the graph. For each variable x_i there is a node for the lower bound ℓ_i, $x^{(\ell_i)}$. All lower bound nodes are connected to a dummy source node, s, with infinite capacity arcs. This guarantees that the lower bound nodes are always in the source set of any finite cut. In fact all these nodes can be shrunk with the source and not be present explicitly in the graph. We refer to their presence only to facilitate the presentation.

There is a chain of infinite capacity arcs, each going from $x_i^{(k+1)}$ to $x_i^{(k)}$. We refer to these collections of arcs for each x_i as the x_i-chain. These represent the set of inequalities (1) in the sense that any finite cut that contains any node in the source set also contains all of its successors and thus represent a solution satisfying these inequalities.

Each node has weight which is the objective coefficient associated with its corresponding variable. These weights are permitted to be either positive or negative. The lower bound nodes assume the value $w_i \ell_i$, and every other node in the chain has the weight w_i. There are arcs of capacity $|w|$ from s to each node of negative weight w, and arcs of capacity w from each node of positive weight w to t. With those additional arcs, a minimum s,t-cut is shown to correspond to an optimal solution to the monotonized system as described next.

The set of inequalities (3) is represented by arcs of capacity ∞ from node p_i of x_i to node p_j of x_j. These are shown in Figure 2(a).

The network for an inequality involving z is depicted in Figure 2(b). This includes the case of these variables being non binary. For z binary there is only one arc corresponding to z originating at each node in the chain.

The next Lemma is proved in the full version of this paper.

Lemma 5. *The source set of any finite cut in the network corresponds to a feasible integer solution to the system of inequalities.*

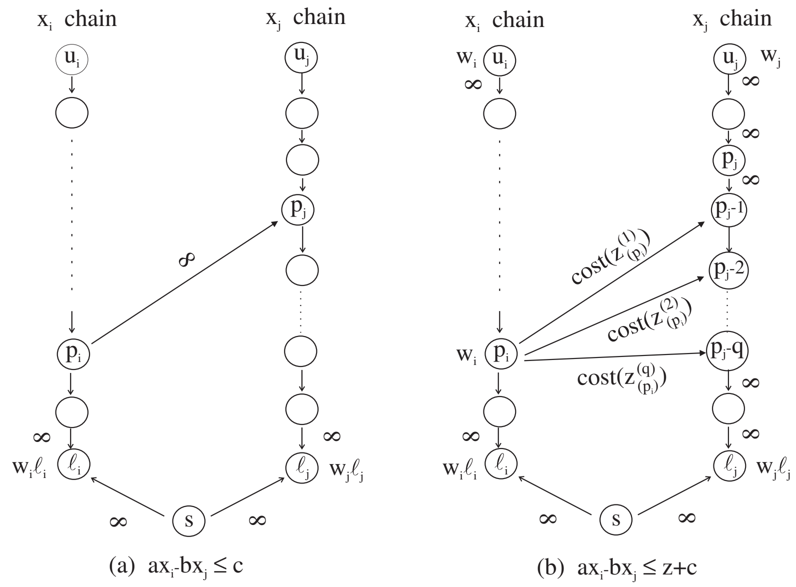

Fig. 2. The network for x_i and x_j

To recover the values of the x and z-variables we set for each node in the source set the corresponding value of 1 to the associated variable, and 0 to those in the sink set. The following equations are then used to recover the values of the x and z variables.

$$x_i = \ell_i + x_i^{(\ell_i+1)} + x_i^{(\ell_i+2)} + \ldots + x_i^{(u_i)},$$

$$z^{(q)} = \sum_{\ell_i \le p_i \le u_i} z_{(p_i)}^{(q)}$$

$$z = \ell_z + \frac{1}{U}[(z(1) - \ell_z)z^{(1)} + (z(2) - z(1))z^{(2)} + \ldots + (z(\bar{q}) - z(\bar{q}-1))z^{(\bar{q})}].$$

Lemma 6. *The minimum s,t-cut in the network constructed corresponds to an optimal solution to the monotonized IP2 with the objective* $\min \sum w_j x_j + \sum c_{ij} z_{ij}$.

Proof. From the previous lemma the source set of a finite cut represents a feasible solution. Consider a finite cut (S,T) in the network, and let the sum of all the negative node weights be W, $W = \sum_{w_j<0}(-w_j)$. Note that W is a constant. Let $C(S,T)$ be the capacity of the cut.

$$\begin{aligned} C(S,T) &= \textstyle\sum_{w_j>0, j\in S} w_j + \sum_{w_i<0, i\in T}(-w_i) + \sum_{i\in S, j\in T} c_{ij} \\ &= \textstyle\sum_{w_i<0}(-w_i) - \sum_{w_i<0, i\in S}(-w_i) + \sum_{w_j>0, j\in S} w_j + \sum_{i\in S, j\in T} c_{ij} \\ &= W + \textstyle\sum_{j\in S} w_j + \sum_{i\in S, j\in T} c_{ij}. \end{aligned}$$

Hence minimizing the cut capacity is equivalent to minimizing the sum of weights of nodes in S plus the capacity of the arcs in the cut separating S from T, which is precisely the value of our objective function. □

3 Minimum Capacity Feasible Cut

The *feasible cut* problem was introduced by Yu and Cheriyan [YC95]. In this problem we are given an undirected graph $G = (V, E)$ with nonnegative edge weights, c_{ij}, k pairs of "commodities" vertices $\{s_1, t_1; \ldots; s_k, t_k\}$ and a specified node $v*$. The problem is to find a partition of V, $(S, \bar{S})$, so that $v* \in S$ and so that for every commodity pair s_ℓ, t_ℓ, has at most one node in S, and such that the cost of the cut $C(S, \bar{S})$ is minimum. Yu and Cheriyan proved that the problem is NP-hard and gave a 2-approximation algorithm. Their algorithm requires to solve a linear program that, as we show here, has optimal solution consisting of half integrals.

Our treatment of the feasible cut problem is applicable to a generalized form of the problem permitting a directed graph and weighted nodes and k commodity sets of nodes $T_1, \ldots, T_k$ where $|T_\ell| \geq 2$ for $\ell = 1, \ldots, k$. The generalized feasible cut problem is to find a partition of the V in a (directed) edge weighted graph $G = (V, A)$, $(S, \bar{S})$, so that $v* \in S$ and so that every commodity set, T_ℓ, has at most one node in S, and the cost of the cut $C(S, \bar{S})$ with the cost of the nodes in the source set is minimum. The undirected version can be formulated as a directed one by replacing each edge $\{i, j\}$ by a pair of arcs (i, j) and (j, i) of equal cost $c_{ij} = c_{ji}$. Let $x_i = 1$ if $i \in S$ and 0 otherwise.

$$
\text{(Feas-Cut)} \quad
\begin{array}{lll}
\text{Min} & \sum_{(i,j)\in A} c_{ij} z_{ij} + \sum_{j\in V} w_j x_j & \\
\text{subject to} & x_i - x_j \leq z_{ij} & (i,j) \in A \\
& x_{p_\ell} + x_{q_\ell} \leq 1 & p_\ell, q_\ell \in T_\ell \text{ for } \ell = 1, \ldots, k \\
& x_{v*} = 1 & \\
& x_i, z_j \ \text{ binary for all } i, j. &
\end{array}
$$

The constraints in Feas-Cut have the 2var structure. To see that the formulation is valid, observe that for an optimal solution $\mathbf{x}^*$ the set $S = \{j : x_j^* = 1\}$ forms the desired cut. Note that $S \neq V$ as required since all vertices in a commodity set but one, must assume the x value 0. We can now monotonize and solve with linear programming. Better still, we construct a network in which a minimum cut where $v*$ is at the source set of an optimal solution. Figure 3 illustrates the basic gadget in the network for feasible cut. In the figure, each inequality in the first set of inequalities corresponds to the two horizontal arcs, and each inequality in the second set corresponds to the diagonal arcs.

After deriving the $\frac{1}{2}$ integral optimal solution $\hat{\mathbf{x}}, \hat{\mathbf{z}}$ we round it to a feasible solution: The values of $\hat{\mathbf{z}}$ that are $\frac{1}{2}$ are all rounded up to 1 and the values of $\hat{\mathbf{x}}$ that are $\frac{1}{2}$ are rounded down to 0. Denote this feasible rounded solution by $\mathbf{x}^*, \mathbf{z}^*$ and the optimal solution value by OPT. This rounded solution is 2-approximate:

$$\sum c_{ij} z_{ij}^* + \sum w_j x_j^* \leq 2 \cdot \sum c_{ij} \hat{z}_{ij} + \sum w_j \hat{x}_j \leq 2 \cdot \text{OPT}.$$

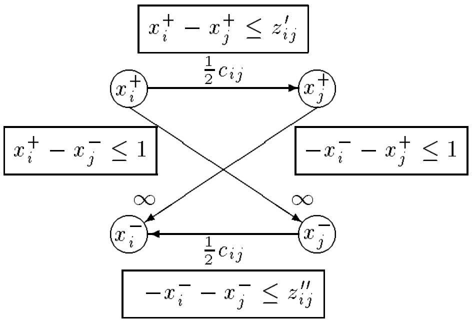

Fig. 3. Feasible cut

As for the complexity, the problem is finding a minimum cut with a specified source on a network with $O(n)$ nodes and $O(M)$ arcs for $M = |A| + \sum^k \binom{|T_\ell|}{2}$. This problem can be solved in the same complexity as a single maximum flow problem in $O(Mn \log \frac{n^2}{M})$ using the algorithm of Hao and Orlin [HO94].

4 Conclusions

Algebraic manipulation of the constraint matrix of some integer programming formulations is shown an effective approximation algorithmic tool. It is hence useful to focus on alternative formulations of problems and perhaps other types of reductions to totally unimodular matrices.

It is intriguing that there are additional half integrality results of [GVY93a] and [GVY94] for multiway directed cuts on edges or nodes (all pairs multicut) and for multicut on trees that we cannot explain with the framework proposed. These might perhaps be explained by reductions to other types of totally unimodular matrices. Another possibility is that there exists another 2var formulation that so far we failed to identify.

There are other problems for which we know of 2-approximations but not of $\frac{1}{2}$ integrality. These include the vertex feedback problem (undirected), the k-center problem and the directed arc feedback set with an objective to maximize the weight of the remaining arcs in the acyclic graph (see [Hoc96a] for descriptions of these problems and approximation algorithms). Are $\frac{1}{2}$ integrality results possible for these problems? At this point the question remains open.

References

[CH97] F. Chudak and D. S. Hochbaum. A half-integral linear programming relaxation for scheduling precedence-constrained jobs on a single machine. Manuscript, UC Berkeley, (1997)

[GVY93a] N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow in trees, with applications to matching and set cover. In *Proceedings of the 20th International Colloquium on Automata, Languages and Programming*, (1993) [GVY94] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in directed and node weighted graphs. In *Proceedings of the 21th International Colloquium on Automata, Languages and Programming*, (1994) 487–498

[GT88] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. *J. of ACM*, **35** (1988) 921–940

[HO94] J. Hao and J. B. Orlin. A faster algorithm for finding the minimum cut in a graph. *Journal of Algorithms* **17** (1994) 424–446

[HN94] D. S. Hochbaum and J. Naor. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM Journal on Computing*, **23** (1994) 1179–1192

[HMNT93] D. S. Hochbaum, N. Megiddo, J. Naor and A. Tamir. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Mathematical Programming*, **62** (1993) 69–83

[HP97] D. S. Hochbaum, A. Pathria. Approximating a Generalization of MAX 2SAT and MIN 2SAT. Manuscript, UC Berkeley, January (1997)

[Hoc82] D. S. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM J. Comput.* **11** (1982) 555-556. An extended version in: W.P. #64-79-80, GSIA, Carnegie-Mellon University, April 1980

[Hoc83] D. S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, **6** (1983) 243–254

[Hoc96] D. S. Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set and related problems. Chapter 3 in *Approximation algorithms for NP-hard problems* edited by D. S. Hochbaum. PWS Boston. (1996)

[Hoc96a] D. S. Hochbaum. Various notions of approximations: Good, Better, Best and more. Chapter 9 in *Approximation algorithms for NP-hard problems* edited by D. S. Hochbaum. PWS Boston. (1996)

[Hoc97] D. S. Hochbaum. Approximating clique and biclique problems. Manuspcript, UC berkeley, June (1997). Revised Jan (1998)

[Hoc97a] D. S. Hochbaum. A framework for half integrality and good approximations. Manuspcript, UC berkeley, June 1996. Revised Sep (1997)

[Hoc98] D. S. Hochbaum. The t- vertex cover problem: Extending the half integrality framework with budget constraints. UC berkeley, manuscript. February (1998). Extended abstract in this volume.

[KKM94] R. Kohli, R. Krishnamurti and P. Mirchandani. The minimum satisfiability problem. *SIAM J. Discrete Mathematics*, **7** (1994) 275–283

[MR96] M. V. Marathe and S. S. Ravi. On approximation algorithms for the minimum satisfiability problem. Manuscript. Jan 1996. To appear *Information Processing Letters.*

[Orl93] J. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations Research* **41** (1993) 338–350

[YC95] B. Yu and J. Cheriyan. Approximation algorithms for feasible cut and multicut problems. *Proc. Algorithms - ESA'95* P. Spirakis (ed.) LNCS 979, Springer, New York, (1995) 394–408