Algorithmica (2004) 39: 189–208
DOI: 10.1007/s00453-004-1085-2

Algorithmica
© 2004 Springer-Verlag New York, LLC

# A Cut-Based Algorithm for the Nonlinear Dual of the Minimum Cost Network Flow Problem[1]

Ravindra K. Ahuja,[2] Dorit S. Hochbaum,[3] and James B. Orlin[4]

**Abstract.** We consider a convex, or nonlinear, separable minimization problem with constraints that are dual to the minimum cost network flow problem. We show how to reduce this problem to a polynomial number of minimum $s$,$t$-cut problems. The solution of the reduced problem utilizes the technique for solving integer programs on monotone inequalities in three variables, and a so-called proximity-scaling technique that reduces a convex problem to its linear objective counterpart. The problem is solved in this case in a logarithmic number of calls, $O(\log U)$, to a minimum cut procedure, where $U$ is the range of the variables. For a convex problem on $n$ variables the minimum cut is solved on a graph with $O(n^2)$ nodes. Among the consequences of this result is a new cut-based scaling algorithm for the minimum cost network flow problem. When the objective function is an arbitrary nonlinear function we demonstrate that this constrained problem is solved in pseudopolynomial time by applying a minimum cut procedure to a graph on $O(nU)$ nodes.

**Key Words.** Nonlinear integer programming, Convex integer programming, Total unimodularity, Minimum cut, Network flow.

## 1. Introduction.

We consider a convex, or nonlinear, separable minimization problem with constraints that are dual to the minimum cost flow problem. We show how to reduce this problem to a polynomial number of minimum $s$,$t$-cut problems. The problem's constraints are of the form $x_i - x_j \leq c_{ij} + z_{ij}$ and their coefficients form a totally unimodular matrix. Convex optimization problems over constraints of this type have varied applications. One extensively studied application area is the problem of statistical estimation subject to rank order constraints, see [BBBB]. Other applications, described in a recent article by Ahuja et al. [AHO], include the time–cost tradeoff in project scheduling, just in time scheduling, inverse spanning tree, and dial-a-ride transit problem. An application to the problem of multi-echelon production lot sizing is described in [HQ]. Another application for the image segmentation problem is detailed in [H2].

With a linear objective function, the problem we study is the dual of the minimum cost network flow problem, as shown next. We therefore refer to the problem studied here as (Dual).

[1] The first author's research was supported in part by NSF Award No. DMI-0217359, that by the second author was supported in part by NSF Awards Nos. DMI-0085690 and DMI-0084857 and by a UC-SMART award, and the third author's research was supported in part by NSF Grant DMI-9820998 and ONR Grant N00014-98-1-0317.
[2] Industrial & Systems Engineering, University of Florida, Gainesville, FL 32611, USA. ahuja@ufl.edu.
[3] Industrial Engineering & Operations Research and Walter A. Haas School of Business, University of California, Berkeley, CA 94720, USA. dorit@hochbaum.ieor.berkeley.edu.
[4] Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. jorlin@mit.edu.

Received March 5, 2002; revised January 15, 2003. Communicated by H. N. Gabow.
Online publication February 25, 2004.

The general formulation of the problem addressed here has each structural constraint involving a pair of variables and possibly a third variable that appears in that constraint only. Let the set of constraints with three variables be $E_1$ and the set of constraints with two variables be $E_2$. Let the set of variables be $V = \{1, \ldots, n\}$. The set of constraints $E$ is partitioned into two subsets $E = E_1 \cup E_2$, with $|E_1| = m_1$, $|E_2| = m_2$, $|V| = n$, and $|E| = m_1 + m_2 = m$. Let the functions $e_{ij}()$ be convex. The primary problem we address in this paper is

$$\text{(Dual)}\qquad \begin{array}{ll} \text{Min} & \displaystyle\sum_{j=1}^{n} w_j(x_j) + \sum_{(i,j)\in E_1} e_{ij}(z_{ij}) \\ \text{subject to} & x_i - x_j \le c_{ij} + z_{ij} \quad \text{for} \quad (i,j) \in E_1, \\ & x_i - x_j \le c_{ij} \quad \text{for} \quad (i,j) \in E_2, \\ & \ell_j \le x_j \le u_j, \quad j = 1, \ldots, n, \\ & \beta_{ij} \le z_{ij} \le \gamma_{ij} \quad \text{for} \quad (i,j) \in E_1, \\ & x_j \text{ integer} \quad \text{for all} \quad j = 1, \ldots, n, \\ & z_{ij} \text{ integer} \quad \text{for all} \quad (i,j) \in E_1. \end{array}$$

When the functions $w_j()$ are convex we call the problem (Convex Dual). Note that the coefficients of the constraints form a totally unimodular matrix, and thus every subdeterminant of this matrix is of value 1, $-1$, or 0.

Without loss of generality the lower bound constraints on the variables $z_{ij}$ may be replaced by 0 lower bound, $0 \le z_{ij} \le \gamma'_{ij}$. Also without loss of generality, all the parameters $\ell_j, u_j, \beta_{ij}, \gamma_{ij}$, and $c_{ij}$ are integers.

When the objective function of (Dual) is linear, $w_j(x_j) = w_j x_j$ and $e_{ij}(z_{ij}) = \bar{u}_{ij} z_{ij}$, we refer to the problem as (Linear Dual). The dual of (Linear Dual) is the minimum cost network flow problem (Flow):

$$\text{(Flow)}\qquad \begin{array}{ll} \text{Min} & \displaystyle\sum_{ij\in E_1\cup E_2} c_{ij} y_{ij} + \sum_{i=1}^{n} u_i \alpha_i + \sum_{ij\in E_1} \gamma_{ij}\delta_{ij} \\ \text{subject to} & \displaystyle -\sum_k y_{ik} + \sum_k y_{ki} - \alpha_i \le w_i, \qquad i \in V, \\ & \bar{u}_{ij} \ge y_{ij} - \delta_{ij} \ge 0, \quad (i,j) \in E_1, \\ & y_{ij} \ge 0, \quad (i,j) \in E_1 \cup E_2, \\ & \delta_{ij} \ge 0, \quad (i,j) \in E_1, \\ & \alpha_i \ge 0, \quad i = 1, \ldots, n. \end{array}$$

(Flow) is the formulation of a network flow problem on a network with $n$ nodes—one per structural constraint, and a dummy node, $r$, serving as a root. The variable $\alpha_i$ represents the flow from node $i$ to the root. The inflow to node $i$ exceeds the outflow by at most $w_i$. This quantity is assigned as capacity to arcs going from node $i$ to the root. The costs of these arcs are $u_i$. The costs of all other arcs not adjacent to root is $c_{ij}$. For each such arc that belongs to $E_1$ there is an additional, parallel, arc of unbounded capacity with a cost of $c_{ij} + \gamma_{ij}$. The amount of flow on this parallel arc is $\delta_{ij}$ and this flow is positive only if the flow on the first arc has reached its capacity $\bar{u}_{ij}$.

Once the minimum cost flow problem is solved, the dual variables—the *potentials*—are derived using complementary slackness and shortest paths procedure. For details on the procedure the reader is referred to the book by Ahuja et al. [AMO] (page 316).

Therefore (Linear Dual) can be solved in the same running time as a minimum cost network flow.

Our approach here is distinguished in that it does not solve the flow problem in any way. All the routines are in the (Dual) space and utilize only a minimum cut procedure.

1.1. *Prior Research.* The convex optimization over linear constraints problem was known, since the early 1950s, to be solved via linear programming by replacing each variable $x_j$ by a sum of binary variables $\sum_{k=\ell_j}^{u_j} x_j^{(k)}$ (see, e.g., [D]). The resulting linear program produces an optimal solution to the convex problem in pseudopolynomial time. The running time is pseudopolynomial because it is a polynomial function of the parameter $U = \max_j\{u_j - \ell_j\}$.

The problem (Convex Dual) was first shown to be solvable in polynomial time by Hochbaum and Shanthikumar [HS]. It was shown there that any convex separable optimization over linear constraints is solved in $\log U$ calls to a linearized version of the problem in $O(n^2\Delta)$ variables where $\Delta$ is the largest subdeterminant of the constraint matrix and $U$ is a bound on the range of the variables. Specifically, any convex separable minimization in integers over a totally unimodular constraint matrix was shown to be solvable in polynomial time by $O(\log U)$ calls to a linearized version of the problem in binary variables. One of our contributions here is to show that this linearized version is solvable by a minimum cut procedure.

The problem of convex optimization over totally unimodular constraints was also addressed by Karzanov and McCormick [KM]. They proposed two algorithms for the problem, one based on a minimum mean cycle canceling method and the other based on a "cancel-and-tighten" approach. Specialized to our problem, called the *network cocirculation*, the run times for these methods are $O(m^2n^2\log^2(n\max\{U, C\}))$ and $O(n\log(n\max\{U, C\})(m + n\log n))$, respectively, for $C = \max_{ij} c_{ij}$.

In [AHO] we presented a polynomial time algorithm for (Convex Dual) which is the most efficient algorithm known to date for the problem. The complexity of the algorithm is $O(mn\log n\log(nC))$. The algorithm uses Lagrangian relaxation to recast the problem as a convex cost network flow problem. Certain features of that convex cost network flow problem make it possible to apply a particularly efficient algorithm based on the successive approximation technique of Goldberg and Tarjan [GT2]. Our algorithm here, by comparison, solves the (Convex Dual) problem directly, without dualizing it first.

A special case of (Convex Dual) has been addressed recently by Hochbaum and Queyranne, [HQ]. In that problem, which we refer to as the *convex closure problem*, the constraints are of the form $x_i - x_j \leq 0$. The algorithm reported has the complexity of a single minimum cut procedure, $O(mn\log(n^2/m))$, plus the work required to find the integer minima of the $n$ convex functions in the objective, $O(n\log U)$.

1.2. *The Main Contributions.* The algorithm we develop here for (Dual) solves the problem by constructing a graph associated with the problem and solving a minimum cut problem on that graph. When the graph corresponding to (Dual) has a special structure it is easy to exploit this structure in the minimum cut procedure. For example, the inverse spanning tree problem is an application in which this special structure is advantageous [H5]. For the objective of minimizing the sum of absolute deviations, the graph structure resembles that of a bipartite simple network, which leads to an algorithm with the best run time for the problem [H5]. Another advantage is practical—in order to solve (Dual) one requires nothing more sophisticated than a minimum cut routine, which is more readily available than convex optimization software.

We describe an application of the cut-based algorithm to two different scenarios of the problem. In one we have arbitrary nonlinear functions $w_j()$ and the running time is pseudopolynomial. When the functions $w_j()$ are convex the running time is polynomial. We also consider a special case of the convex problem where $E_2 = \emptyset$ in which there is an improvement in the complexity of the polynomial time algorithm.

Let $T(n, m)$ denote the run time required to solve a minimum cut problem on a graph with $n$ nodes and $m$ arcs. Recall that throughout, unless otherwise stated, $e_{ij}()$ are convex functions.

1. When the functions $w_j()$ are general *nonlinear* functions, the running time of our algorithm is pseudopolynomial, $T(nU, mU^2)$.
2. For $w_j()$, $e_{ij}()$ convex functions, the running time of the algorithm is $O(\log U \cdot T(n^2, mn^2))$. If the functions $e_{ij}()$ are linear or piecewise linear with a fixed number of pieces the running time is $O(\log U \cdot T(n^2, mn))$. In the linear case this leads to a new polynomial time algorithm for minimum cost network flow that solves a sequence of minimum cut problems, followed by a single maximum flow problem.
3. When $w_j()$ are convex, there are no variables $z_{ij}$ ($E_2 = \emptyset$), and $\ell$ is largest so that $c_{ij} = k_{ij}2^\ell$ for $k_{ij}$ integers (or 0), then the problem is solved in time $O(\min\{T(n(U/2^\ell), m(U/2^\ell)), \log(U/2^\ell) \cdot T(n^2, nm)\} + \ell T(n, m))$. (Note that $c_{ij} < U$ and thus $\ell \le \log U$.)

Our algorithm for the general nonlinear case is based on a technique of integer optimization over *monotone* constraints with up to three variables in each constraint [H4]. Monotone inequalities have up to two variables appearing with opposite signs coefficients such as $ax - by \le c + z$ for $a$, $b$ nonnegative, and a third variable $z$ appearing in one constraint at most. (Dual) is obviously such a problem. The algorithm for integer optimization over monotone constraints runs in time that depends on the range of the variables $x$ and $y$ (which is $U$ for (Dual)).

In the convex case the run time of our algorithm is improved to polynomial run time based on the proximity-scaling algorithm of Hochbaum and Shanthikumar [HS]. That generic approach works as follows: First we scale the units of the variables to a scaling unit of size $s$, then we replace the variables by the sum of binary variables. The number of binary variables replacing each variable is $O(u_j/s)$. The resulting piecewise linearized problem is solved using the algorithm for optimizing over monotone inequalities. The scaling unit $s$ is selected so that $u_j/s$ is a polynomial quantity. The procedure then relies on a proximity theorem that guarantees that the scaled problem solution is close enough to the optimum so that the range of the variables can be reduced by a constant factor.

For the case when there are no $z_{ij}$ variables we employ a strong proximity theorem, proved in this paper, which demonstrates that for right-hand sides that are integer multiples of the scaling factor, the optimal solution to the scaled problem lies within two scaling units away from an optimal solution at the previous scaling phase.

The presentation here is organized as follows: Section 2 presents a pseudopolynomial algorithm for the problem that is applicable to the nonconvex problem as well as the convex problem. In Section 3 we describe the proximity-scaling algorithm of Hochbaum and Shanthikumar and how it applies to the (Convex Dual) problem calling $\log U$ times to the minimum cut procedure of Section 2, each with (strongly) polynomial time complexity. Next we prove a stronger version of the proximity theorem in Section 4 which is restricted to the (Convex Dual) problem in two variables per inequality with right-hand sides that are divisible by the scaling factor. This proximity theorem demonstrates that the optimal solution to the scaled problem is within two scaled units away from the optimal solution to the subsequent scaled problem and leads to improved run times for (Convex Dual) problems satisfying the required conditions. In Section 5 it is shown how to use the proximity-scaling algorithm to solve the minimum cost network flow problem using a logarithmic number of calls to a minimum cut procedure and one call to a maximum flow procedure.

1.3. *Notation*. We denote vectors by bold characters, e.g., $\mathbf{x}$. The vector $\mathbf{e}$ is the array of ones $(1, \ldots, 1)$.

Let $G = (V, A)$ be a connected directed graph on $n$ nodes and $m$ arcs. Let $c_{ij}$ be the capacity of arc $(i, j)$ in the graph. A partition of $V$ is a cut $(B, \bar{B})$ of capacity $\sum_{i\in B j\in \bar{B}} c_{ij}$. We consider $s$,$t$-cuts that are cuts in a graph that contains distinguished source $s$ and sink $t$. An $s$,$t$-cut is a partition $(B, \bar{B})$ so that $s \in B$ and $t \in \bar{B}$. The minimum $s$,$t$-cut problem is to find an $s$,$t$-cut of minimum capacity. In this paper we refer to the minimum $s$,$t$-cut problem as the minimum cut problem. We call an arc capacitated graph containing distinguished source and sink nodes, an $s$,$t$-graph

In the complexity expressions we let $T(n, m)$ be the time required to solve a minimum cut problem on a graph $G$ with $n$ nodes and $m$ arcs. The quantity $T(n, m)$ may be assumed to be of order $O(mn \log(n^2/m))$ [GT1], or $O(\min\{n^{2/3}, m^{1/2}\}m \log(n^2/m) \log U)$ [GR], or, in case a randomized algorithm is acceptable, $O(mn + n^2 \log^2 n)$ with probability at least $1 - 2^{-\sqrt{mn}}$ [CH]. $T_{\mathrm{MF}}(n, m)$ is the run time required to solve the maximum flow problem. To date there is no algorithm known for a minimum cut of complexity faster than that of maximum flow. In practice though, maximum flow requires an additional 25–100% run time more than minimum cut as reported in a study by Anderson and Hochbaum of the pseudoflow algorithm of [H1] and the push-relabel algorithm of [GT1].

We denote $U = \max_j\{u_j - \ell_j\}$ and $C = \max_{(i,j)\in E} c_{ij}$.

## 2. A Pseudopolynomial Algorithm.

The problem addressed in this section is (Dual) with the functions $w_j()$ arbitrary nonlinear. The functions $e_{ij}()$, however, are still required to be convex. The pseudopolynomial algorithm presented in this section is the building block for the polynomial time algorithms that apply to the convex instances of the (Dual).

We first provide a description of the algorithm for (Dual) when all constraints are monotone inequalities in two variables. That is, these are constraints of the form $ax - by \leq c$ where both $a$ and $b$ are nonnegative. Later we address the algorithm for the problem with constraints that include monotone inequalities in three variables. These are defined to be of the form $ax - by \leq c + z$ for $a$ and $b$ nonnegative and where the third variable $z$ appears in at most one inequality.

Integer programs on monotone constraints with at most two variables per inequality, are solvable in time polynomial in the problem size and the value of the largest range $U$ by the algorithm of Hochbaum and Naor [HN]. The nonlinearity or convexity of the objective function does not affect the complexity of the algorithm. We illustrate the application of that algorithm to the special case considered here where $a = b = 1$. Integer programs on monotone inequalities with at most *three* variables per inequality are also solvable in pseudopolynomial time. We present a procedure based on minimum cut which was introduced in [H4].

We begin with the construction of the graph and the equivalent formulation in binary variables for the two variables per inequality case. This is followed by the construction of the graph and the binary formulation for the three variables per inequality case.

2.1. *A Reduction of the Two Variable Problem to the Minimum Cut Problem.* The minimum cut problem is formulated in binary variables. The key to the reduction is to cast our problem in binary variables. To that end, each variable, $x_j$, is replaced by $u_j - \ell_j$ binary variables, $x_j = \ell_j + \sum_{p=\ell_j+1}^{u_j} x_j^{(p)}$. The value of $x_j$ is represented by an array of values assigned to the binary variables consisting of a sequence of ones followed by a sequence of zeros, with either sequence possibly empty. Thus $x_j^{(p)} = 1$ if and only if $x_j \geq p$. The objective function is replaced by the linear objective function defined on binary variables, $\min \sum_{j=1}^{n} \sum_{p=\ell_j+1}^{u_j} [w_j(p) - w_j(p-1)] x_j^{(p)}$.

To enforce the contiguity of the sequences the following inequalities must be satisfied for $j = 1, \ldots, n$,

$$x_j^{(p)} \leq x_j^{(p-1)}, \qquad p = \ell_j + 1, \ldots, u_j, \quad x_j^{(\ell_j)} = 1.$$

With these inequalities $x_j^{(p)} = 1$ only if $x_j^{(k)} = 1$ for $\ell + 1 \leq k \leq p - 1$.

Consider a constraint of the type $x_i - x_j \leq c$. To enforce the satisfaction of such a constraint we replace it by a set of up to $U$ constraints on the corresponding binary variables. First we observe that the constraint is equivalent to up to $u_i - \ell_i$ implications of the type:

$$\text{If} \quad x_i \geq p, \quad \text{then} \quad x_j \geq p - c \qquad \text{for} \quad p = \ell_i + 1, \ldots, u_i \quad \text{and} \quad p - c \leq u_j.$$

If $p - c > u_j$, then in any feasible solution $x_i < p$, which means that $x_i^{(p)}, \ldots, x_i^{(u_i)} = 0$. These implications can be written in terms of the binary variables $x_i^{(k)}$ as the set of $u_i - \ell_i$ inequalities which are equivalent to the constraint $x_i - x_j \leq c$,

$$x_i^{(p)} \leq x_j^{(p-c)}, \qquad p = \ell_i + 1, \ldots, u_i.$$

Each binary variable $x_i^{(p)}$ is assigned a weight equal to its incremental contribution to the objective function:

$$\begin{aligned} w_j^{(\ell_j)} &= w_j(\ell_j), \\ w_j^{(p)} &= w_j(p) - w_j(p-1) \qquad \text{for} \quad p = \ell_j + 1, \ldots, u_j. \end{aligned}$$

If (Dual) is feasible, then the following (Binary Dual) is an equivalent formulation to that of (Dual) that has up to two variables per inequality. If (Dual) is infeasible, then either the infeasibility leads to a contradiction diagnosed at the formulation stage—e.g., an implication of the type "$x_i \geq \ell_i$ then $x_j \geq u_j$"—or else (Binary Dual) is infeasible:

$$\text{(Binary Dual)}\quad \begin{aligned} &\text{Min} \sum_{j=1}^{n} \sum_{p=\ell_j+1}^{u_j} w_j^{(p)} x_j^{(p)} \\ &\text{subject to} \quad x_i^{(p)} \leq x_j^{(p-c)} \ \text{ for } (i,j) \in E \ \text{ and } \ p = \ell_i + 1, \ldots, u_i, \\ &\qquad\qquad\quad x_j^{(p)} \leq x_j^{(p-1)} \ \text{ for } j = 1, \ldots, n \ \text{ and } \ p = \ell_j + 1, \ldots, u_j, \\ &\qquad\qquad\quad x_j^{(p)} \text{ binary for } j = 1, \ldots, n \ \text{ and } \ p = \ell_j + 1, \ldots, u_j. \end{aligned}$$

The problem has thus been equivalently restated in binary variables and constraints that each have one 1 coefficient and one $-1$ coefficient. These constraints characterize the minimum closure problem which is solvable by a minimum cut procedure as shown next.

A set of nodes in a directed graph is said to be a *closed set* if it contains all successors of nodes in the set. Given a directed graph $G = (V, A)$ with node weights $w_j$ for all $j \in V$, the *minimum closure* problem is to find a closed set $S \subseteq V$ such that $\sum_{j \in S} w_j$ is minimized.

The graph construction associated with (Binary Dual) is as follows: Each binary variable has a node corresponding to it in the graph with a weight equal to the weight coefficient in the objective function of (Binary Dual). For each constraint $x \leq y$ there is an arc leading from the node corresponding to $x$ to the node corresponding to $y$. In particular, the graph contains an $x_j$*-chain* associated with each variable $x_j$ consisting of a sequence of arcs from each node representing $x_j^{(p)}$ to the node representing $x_j^{(p-1)}$ for $p = \ell_j + 1, \ldots, u_j$. Let a variable be 1 if and only if the corresponding node is in the closure. Then the set of nodes valued 1 is closed under succession in this graph if and only if the corresponding assigned values to the variables satisfy all constraints. Figure 2.1 describes the construction of the graph for two variables.

The problem is now to find a closed set in the defined graph of minimum total weight, which is precisely the minimum closure problem. The minimum closure problem is

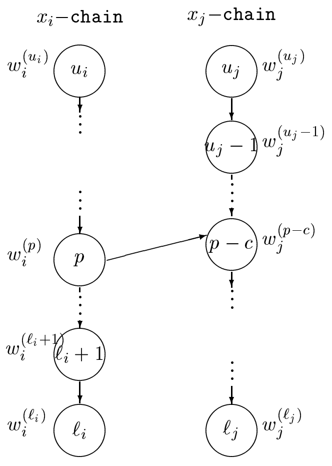

**Fig. 2.1.** The graph representing two variables and an inequality that links them.

solved by finding a minimum $s$,$t$-cut in an associated arc-capacitated $s$,$t$-graph as was first demonstrated by Picard [P]. We show how this is done for (Binary Dual).

In order to solve the closure problem associated with (Binary Dual) as a minimum cut problem we add a source $s$ and a sink $t$ to the graph. Every node $i$ in the graph that has a positive weight $w_i > 0$ is connected to the sink $t$ with an arc of capacity $w_i > 0$. The source $s$ is connected to every node $i$, that has negative weight $w_i$, with an arc of capacity $|w_i| = -w_i > 0$. The source set of the resulting minimum $s$,$t$-cut corresponds, as we show next, to the optimal solution to the problem by setting the value of each binary variable that has a corresponding node in the source set to 1. All other arcs, that are not adjacent to a source or sink, assume infinite capacity.

Let $(S, \bar{S})$ be a finite cut and let the solution to (Dual) $x_i = \ell_i + \sum_{p=\ell_i+1}^{u_i} x_i^{(p)}$ associated with the cut derived from the setting be

$$x_i^{(p)} = \begin{cases} 1 & \text{if node } x_i^{(p)} \in S, \\ 0 & \text{if node } x_i^{(p)} \in \bar{S}. \end{cases}$$

In order to enforce the nodes $x_i^{(\ell_i)}$ to belong to the source set we have arcs of infinite capacity from the source to every node $x_i^{(\ell_i)}$. (These nodes can alternatively be shrunk with the source and eliminated from their respective chains.)

LEMMA 2.1. *The solution* **x** *associated with a minimum cut* $(S, \bar{S})$ *is a feasible and optimal solution to the* (*Binary Dual*) *problem.*

PROOF. A generic constraint $x \leq y$ of (Binary Dual) is violated only if $x = 1$ and $y = 0$. However, such an assignment corresponds to an infinite capacity arc going from $S$ to $\bar{S}$ and thus violates the finiteness of the cut. Indeed, the source set of *any* finite cut corresponds to a closed set, else it contains an infinite capacity arc violating finiteness.

Let $V^+$ and $V^-$ represent the set of nodes with positive and negative weight, respectively. Let $(S, \bar{S})$ be a finite $s$,$t$-cut with $s \in S$ and $t \in \bar{S}$. The arcs in the cut are either in $(S \cap V^+, \{t\})$ or in $(\{s\}, \bar{S} \cap V^-)$. Denote the sum of weights of nodes in a set $D$ by $w(D) = \sum_{i \in D} w_i$

$$\begin{aligned} C(S, \bar{S}) &= \sum_{i \in \bar{S} \cap V^-} (-w_i) + \sum_{i \in S \cap V^+} w_i \\ &= \sum_{i \in V^-} (-w_i) - \sum_{i \in S \cap V^-} (-w_i) + \sum_{i \in S \cap V^+} w_i = -w(V^-) + \sum_{i \in S} w_i. \end{aligned}$$

The first term $-w(V^-)$ in the last expression is a constant, the sum of all negative weight nodes. The second term is the weight of the nodes in the source set. Thus the capacity of the cut and the weight of the closed set differ by a constant. In particular, the source set of a minimum cut corresponds to a minimum value closed set, as required. □

In order to solve the problem (Binary Dual) it is thus only necessary to find a minimum cut in the graph constructed and set all the binary variables in the source set of the cut to the value 1, and those in the sink set to the value 0. The complexity of solving the problem is therefore the complexity of finding a minimum cut in the graph associated with (Binary Dual), that has $O(nU)$ nodes and $O(mU)$ arcs, $T(nU, mU)$.

2.2. *A Reduction of the Three Variables Problem to a Minimum Cut Problem.* We first provide a binary formulation equivalent to (Dual), here called (BD), and then show an associated graph construction where the solution to the minimum cut problem provides a solution to (Dual) with the three variables constraints. The formulation includes all inequalities associated with the two variables constraints, and the graph includes the same construction of the $x_i$-chains and the associated finite capacity arcs that connect the nodes to source or sink as in the two variables case. Both (BD) and the graph are created by appending the (Binary Dual) problem for the two variables inequalities, and appending the corresponding $s$,$t$-graph, with a set of inequalities and arcs that represent the three variable inequalities. Each appended inequality is of the form $x_i^{(p)} \le x_j^{(q)} + z_{ij}(p,q)$ where $z_{ij}(p,q)$ is a binary variable. The binary variables $z_{ij}(p,q)$ are each associated with an arc in the graph from node $x_i^{(p)}$ to node $x_j^{(q)}$.

In order to simplify the presentation we introduce a sequence of assumptions that hold without loss of generality:

1. $0 \le z_{ij} \le \gamma_{ij}$. The lower bound, if nonzero, can then be added to the right-hand side constant $c_{ij}$.
2. $e_{ij}(0) = 0$, or else add a constant to the objective function.
3. The functions $e_{ij}()$ are assumed to be nondecreasing. Else set $z_{ij}$ to be at least $z_{ij}^*$, where $e_{ij}()$ is nondecreasing for $z_{ij} \ge z_{ij}^*$. The value $z_{ij}^*$ is the argument at which the convex function $e_{ij}()$ attains its minimum.
4. Since $e_{ij}()$ are nondecreasing, the value of $z_{ij}$ is determined as
$$z_{ij} = \max\{0, x_i - x_j - c_{ij}\}.$$
5. The functions $e_{ij}()$ are extended to nondecreasing convex functions on the real line by setting, for a suitably large value of $M$,
$$e_{ij}(z) = \begin{cases} 0 & \text{if} \quad z < 0, \\ e_{ij}(z) & \text{if} \quad 0 \le z \le \gamma_{ij}, \\ e_{ij}(\gamma_{ij}) + M(z - \gamma_{ij}) & \text{if} \quad z > \gamma_{ij}. \end{cases}$$
6. The variables $z_{ij}$ assume values on the entire real line and $z_{ij} = x_i - x_j - c_{ij}$.

When done with the process of generating the inequalities, to be described next, the resulting binary dual (BD) is

$$\text{(BD)} \quad \begin{aligned} &\text{Min} \sum_{j=1}^{n} \sum_{p=\ell_j+1}^{u_j} w_j^{(p)} x_j^{(p)} + \sum_{(i,j)\in E_1} \sum_{p=\ell_i+1}^{u_i} \sum_{q=\ell_j+1}^{u_j} c_{ij}(p,q) z_{ij}(p,q) \\ &\text{subject to} \quad x_i^{(p)} \le x_j^{(p-c_{ij})} \quad \text{for} \quad (i,j) \in E_2 \quad \text{and} \quad p = \ell_i + 1, \ldots, u_i, \\ &\qquad x_i^{(p)} \le x_j^{(q)} + z_{ij}(p,q) \\ &\qquad\quad \text{for} \quad (i,j) \in E_1 \quad \text{and} \quad p = \ell_i + 1, \ldots, u_i, \\ &\qquad\quad q = p - c_{ij} - \gamma_{ij}, \ldots, p - c_{ij} - 1, \\ &\qquad x_j^{(p)} \le x_j^{(p-1)} \quad \text{for} \quad j = 1, \ldots, n \quad \text{and} \quad p = \ell_j + 1, \ldots, u_j, \\ &\qquad x_j^{(p)} \text{ binary} \quad \text{for} \quad j = 1, \ldots, n \quad \text{and} \quad p = \ell_j + 1, \ldots, u_j, \\ &\qquad z_{ij}(p,q) \text{ binary} \quad \text{for} \quad (i,j) \in E_1 \quad \text{and} \quad p = \ell_i + 1, \ldots, u_i, \\ &\qquad\quad q = \ell_j + 1, \ldots, u_j. \end{aligned}$$

Consider a constraint with three variables, $x_i - x_j \le c_{ij} + z_{ij}$. Since the functions $e_{ij}()$ are convex and nondecreasing,

$$e_{ij}(x+1) - e_{ij}(x) \ge e_{ij}(x) - e_{ij}(x-1). \tag{2.1}$$

We define the excess unit increments of the functions $e_{ij}$ (a discrete equivalent of the second derivative):

$$\Delta_{ij}(k) = [e_{ij}(k) - e_{ij}(k-1)] - [e_{ij}(k-1) - e_{ij}(k-2)].$$

The functions $\Delta_{ij}()$ are nonnegative as follows from the convexity in (2.1). Note that when the function $e_{ij}()$ is linear then only $\Delta_{ij}(1)$ is nonzero, and when $e_{ij}()$ is piecewise linear then the number of arguments for which $\Delta_{ij}()$ is nonzero is equal to the number of "pieces" of $e_{ij}()$.

We now proceed in two steps, first regarding the variables $x_i$, $x_j$ as unbounded, and then treating the bounded case.

*Unbounded Variables $x_i$.* We deal with the case $\ell_i = -\infty$ and $u_i = +\infty$ for all $i$. We model the $x_i$ and $z_{ij}$ as sums of binary variables. We have binary variables $x_i^{(p)}$, $p \in \mathbb{N}$, with

$$x_i^{(p)} = 1 \qquad \text{iff} \quad p \le x_i,$$

and binary variables $z_{ij}(p,q)$ for $p, q \in \mathbb{N}$ with

$$z_{ij}(p,q) = 1 \qquad \text{iff} \quad p \le x_i, \quad q > x_j, \quad \text{and} \quad p - q \ge c_{ij}.$$

This is equivalent to the inequality $x_i^{(p)} - x_j^{(q)} \le z_{ij}(p,q)$. We next modify the objective function. For the part depending on the $x_i$ we proceed as in the previous section. We have

$$w(x_i) = \sum_p (w(p) - w(p-1)) x_i^{(p)} = \sum_p w_i^{(p)} x_i^{(p)}.$$

For $\sum_{ij} e_{ij}(z_{ij})$ we have to work slightly harder. For

$$\Delta_{ij}(l) = (e_{ij}(l) - e_{ij}(l-1)) - (e_{ij}(l-1) - e_{ij}(l-2)),$$

we observe that $\Delta_{ij}(l) = 0$ for $l \le 0$ and recall that $\Delta_{ij}(l) \ge 0$ for all $l$. Then we claim that

$$e_{ij}(z_{ij}) = \sum_{pq} \Delta_{ij}(p - q - c_{ij} + 1) z_{ij}(p,q).$$

This can be seen as follows (we drop the subscript $ij$ for simplicity):

$$\begin{aligned}
\sum_{pq} \Delta(p - q - c + 1) z(p,q) &= \sum_{p \le x_i,\ q > x_j} \Delta(p - q - c + 1) \\
&= \sum_{p \le x_i,\ -q < -x_j} \Delta(p - q - c + 1) \\
&= \sum_{p \le x_i,\ q < -x_j} \Delta(p + q - c + 1)
\end{aligned}$$

$$\begin{aligned}
&= \sum_{p \le x_i,\ q \le -x_j} \Delta(p+q-c) \\
&= \sum_{p \le x_i} e(p - x_j - c) - e(p - x_j - c - 1) \\
&= e(x_i - x_j - c) - e(0) \\
&= e(z).
\end{aligned}$$

The contiguity conditions on the variables are easily formulated as before,

$$x_i^{(p)} \ge x_i^{(p-1)} \qquad \text{for all } p,$$

$$\sum_p x_i^{(p)} \ge 1, \qquad \sum_p (1 - x_i^{(p)}) \ge 1,$$

where the last two inequalities guarantee that some but not all $x_i^{(p)}$ are one. We also have

$$x_i^{(p)} \le x_j^{(q)} + z_{ij}(p,q) \qquad \text{for} \quad p, q \in \mathbb{N}.$$

The last set of inequalities captures the if-part in the definition of $z_{ij}(p,q)$. The only-if part is captured in the objective function using the nonnegativity, $\Delta_{ij}(l) \ge 0$ for all $l$.

*Bounded Variables.* A range restriction $\ell_i \le x_i \le u_i$ is easily modeled by the additional inequalities $x_i^{(p)} = 1$ for $p \le \ell_i$ and $x_i^{(p)} = 0$ for $p > u_i$. Of course, we may remove the fixed variables from the problem. We can also remove many of the $z$-variables. Observe that

$$z_{ij}(p,q) = \begin{cases} 0 & \text{if} \quad p > u_i \quad \text{or} \quad q \le \ell_j, \\ 1 & \text{if} \quad p \le \ell_i \quad \text{and} \quad q > u_j, \\ z_{ij}(\ell_i, q) & \text{if} \quad p < \ell_i, \\ z_{ij}(p, u_j + 1) & \text{if} \quad q > u_j. \end{cases}$$

We may therefore simplify the objective function. The variables fixed to zero contribute zero, the variables fixed to one contribute a constant, and the variables with a common value allow us to combine terms. Let $\ell_j \le q \le u_j$. Then, dropping the subscripts for convenience,

$$\begin{aligned}
\sum_{p \le \ell_i} \Delta(p - q - c + 1) z(p,q) &= \sum_{p \le \ell_i} \Delta(p - q - c + 1) z(\ell_i, q) \\
&= (e(\ell_i - q - c + 1) - e(\ell_i - q - c)) z(\ell_i, q),
\end{aligned}$$

and for $\ell_i < p \le u_i$, we have

$$\begin{aligned}
\sum_{q \ge u_j + 1} \Delta(p - q - c + 1) z(p,q) &= \sum_{q \ge u_j + 1} \Delta(p - q - c + 1) z(p, u_j + 1) \\
&= \sum_{-q \le -(u_j+1)} \Delta(p - q - c + 1) z(p, u_j + 1) \\
&= \sum_{q \le -(u_j+1)} \Delta(p + q - c + 1) z(p, u_j + 1) \\
&= \sum_{q \le -u_j} \Delta(p + q - c) z(p, u_j + 1) \\
&= (e(p - u_j - c) - e(p - u_j - c - 1)) z(\ell_i, q).
\end{aligned}$$

This establishes the validity of the formulation (BD) as equivalent to (Dual).

REMARK 2.1. If the functions $e_{ij}()$ are linear, then per inequality $x_i - x_j \leq c_{ij} + z_{ij}$ and a value of $x_i = p$ there is at most one variable $z_{ij}(p, q)$ which is not fixed at zero. If the functions $e_{ij}()$ are piecewise linear with at most $k$ pieces, then there are at most $k$ variables $z_{ij}(p, q)$ which are not fixed at zero.

REMARK 2.2. The problem formulated as (BD) is an instance of the $s$-excess problem defined as the problem solved by the pseudoflow algorithm of [H1]. There it is shown how the $s$-excess problem generalizes the minimum closure problem and how solving it is equivalent to solving a minimum cut problem.

We now complete the graph construction in which the minimum cut problem solves the problem (BD). Let an arc going from node $x_i^{(k_i)}$ to node $x_j^{(k_j)}$ be the ordered pair $(k_i, k_j)$. For every variable $z(k_i, k_j)$ the corresponding arc $(k_i, k_j)$ goes from node $x_i^{(k_i)}$ to node $x_j^{(k_j)}$. The capacity assigned to arc $(k_i, k_j)$ is the coefficient of the variable $z(k_i, k_j)$ in the objective function of (BD).

Consider a finite cut $(S, \bar{S})$ in the generated graph $G^{\mathrm{BD}} = (V, A)$ (BD stands for Binary Dual), where $V$ is a set of nodes corresponding to the range of values of each variable $x_i$, and a source and sink node, $A$ is a set of arcs consisting of infinite capacity arcs within each $x_i$-chain and corresponding to two variables inequalities, finite capacity arcs between the nodes and source and sink (depending on whether the weight of the node is positive or negative), and arcs corresponding to each variable $z(k_i, k_j)$. We derive a solution corresponding to the minimum cut by letting a variable $x_i^{(p)} = 1$ if the node corresponding to that variable is in the source set of the cut $S$, and 0 otherwise:

$$x_i^{(p)} = \begin{cases} 1 & \text{if} \quad \text{node } x_i^{(p)} \in S, \\ 0 & \text{if} \quad \text{node } x_i^{(p)} \in \bar{S}. \end{cases}$$

We let $z_{ij}(p_1, p_2) = 1$ iff arc $(p_1, p_2)$ is in the cut. We let $z_{ij}$ be equal to the number of nodes in the $x_j$-chain that have at least one cut-arc coming from the $x_i$-chain adjacent to them.

We claim in the next theorem that such a solution is feasible for the problem and that the objective value of this solution is equal to the capacity of the cut plus a fixed constant. We further show that any feasible solution corresponds to a finite cut in the graph. This will lead to the conclusion that the solution corresponding to a minimum cut is optimal for our problem.

EXAMPLE. In Figure 2.2 we present an example showing the generated graph for an inequality $x_i - x_j \leq 2 + z_{ij}$ where $\gamma_{ij} = 3$. The illustration shows the arcs associated with the lowest nodes in the $x_i$-chain. Note that the arcs originating at the first node, $\ell_i$, follow a different pattern from that of the other nodes. This is because the higher valued nodes in the chain are guaranteed that when they are in the source set, then so are all

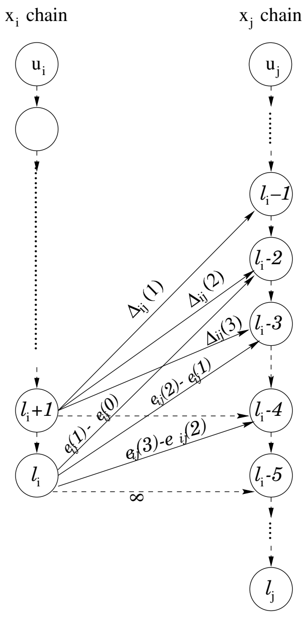

**Fig. 2.2.** The network for $x_i$ and $x_j$.

the nodes under them. Thus the arcs adjacent to a particular valued node in the $x_j$-chain have a corresponding set of arcs originating from lower valued nodes. Each such arc contributes only to the incremental cost of the variable $z_{ij}$.

In the example

$c(\ell_i, \ell_i - 2) = e_{ij}(1) - e_{ij}(0)$. If this arc is in the cut, then $z_{ij} \geq 1$.
$c(\ell_i, \ell_i - 3) = e_{ij}(2) - e_{ij}(1)$. If this arc is in the cut, then $z_{ij} \geq 2$.
$c(\ell_i, \ell_i - 4) = e_{ij}(3) - e_{ij}(2)$. If this arc is in the cut, then $z_{ij} \geq 3$.
$c(\ell_i, \ell_i - 5) = \infty$. It is infeasible for $x_j$ to be $\leq \ell_i - 5$.
$c(\ell_i + 1, \ell_i - 1) = \Delta_{ij}(1) = e_{ij}(1) - e_{ij}(0)$. If this arc is in the cut, then $z_{ij} \geq 1$.
$c(\ell_i + 1, \ell_i - 2) = \Delta_{ij}(2)$. If this arc is in the cut, then the arcs $(\ell_i, \ell_i - 2)$ and $(\ell_i + 1, \ell_i - 1)$ are also in the cut and $z_{ij} \geq 2$. The total capacity of these three arcs is $\Delta_{ij}(1) + \Delta_{ij}(2) + e_{ij}(1) - e_{ij}(0) = e_{ij}(2) - e_{ij}(1) + e_{ij}(1) - e_{ij}(0) = e_{ij}(2) - e_{ij}(0)$ as required.
$c(\ell_i + 1, \ell_i - 3) = \Delta_{ij}(3)$. If this arc is in the cut, then $z_{ij} \geq 3$.
$c(\ell_i + 1, \ell_i - 4) = \infty$. It is infeasible for $x_j \leq \ell_i - 4$ if $x_i \geq \ell_i + 1$.

The following theorem shows that the minimum cut on the constructed graph provides the optimal solution to (BD) and thus to (Dual).

THEOREM 2.1. *A solution to (Dual), $(\mathbf{x}, \mathbf{z})$, is feasible if and only if it corresponds to a finite cut $(S, \bar{S})$ in the graph $G^{\text{BD}}$. Furthermore, the objective value of the solution is equal to the capacity of the cut plus a fixed constant.*

PROOF. Given a feasible solution $(\mathbf{x}, \mathbf{z})$. Let $x_i = p_i$. All nodes in the $x_i$-chain from $\ell_i$ to $p_i$ are assigned to $S$, the source set of the cut (that includes $s$). All nodes corresponding to values in the range $p_i + 1$ to $u_i$ are assigned to a sink set $\bar{S}$.

Consider an inequality involving a pair of variables $x_i = p_i$, $x_j = p_j$, $x_i - x_j \leq c_{ij} + z_{ij}$. If $p_j \geq p_i - c_{ij}$, then there are no arcs associated with this inequality in the cut. Let $p_j = p_i - c_{ij} - q$ for $q \geq 1$. Since the solution is feasible then $q = z_{ij} \leq \gamma_{ij}$. The arcs in the cut are arranged in the list below with each row listing the arcs adjacent to a node in the $x_i$ chain, and each column has the list of arcs adjacent to a node in the $x_j$ chain:

$$\begin{array}{llll}
(p_i, p_i - c_{ij}), & (p_i, p_i - c_{ij} - 1), & \ldots, & (p_i, p_i - c_{ij} - (q-1)), \\
 & (p_i - 1, p_i - 1 - c_{ij}), & \ldots, & (p_i - 1, p_i - 1 - c_{ij} - (q-2)), \\
 & & \vdots & \\
 & & \ldots, & (p_i - (q-1), p_i - c_{ij} - (q-1)).
\end{array}$$

The capacity of the cut arc $(p_i, p_i - c_{ij})$ is $\Delta_{ij}(1)$. The sum of capacities of cut arcs directed into $x_j^{(p_i-1-c_{ij})}$, $(p_i, p_i - c_{ij} - 1)$, and $(p_i - 1, p_i - 1 - c_{ij})$ is $\Delta_{ij}(1) + \Delta_{ij}(2)$. The total sum of the cut arcs between the $x_i$-chain and $x_j$-chain is

$$\begin{aligned}
&\Delta_{ij}(1) + [\Delta_{ij}(1) + \Delta_{ij}(2)] + \cdots + [\Delta_{ij}(1) + \cdots + \Delta_{ij}(q)] \\
&\quad = e_{ij}(1) + [e_{ij}(2) - e_{ij}(1)] + \cdots + [e_{ij}(q) - e_{ij}(q-1)] \\
&\quad = e_{ij}(q).
\end{aligned}$$

If a node $p_i - k = \ell_i$, then the charge of the arcs adjacent to it includes the charges of the arcs that would have been below it. (This explains why the arcs adjacent to $\ell_i$ are charged differently than those adjacent to nodes above it in the $x_i$-chain.) Therefore the cost of the variables $z_{ij}$ is properly charged. Namely, if $z_{ij} = q$, then the charge to the cut capacity is $e_{ij}(q)$.

Let $c(u, v)$ denote the capacity of the arc $(u, v)$. The total capacity of the cut is

$$\begin{aligned}
C(S, \bar{S}) &= \sum_{i \in \bar{S} \cap V^-} (-w_i) + \sum_{i \in S \cap V^+} w_i + \sum_{(i,j) \in E_1} \sum_{k_i = \ell_i}^{p_i} \sum_{k_j = p_j + 1}^{u_j} c(k_i, k_j) \\
&= w(V^-) + \sum_{i \in S} w_i + \sum_{(i,j) \in E_1} \sum_{p_i = \ell_i}^{u_i} \sum_{p_j = \ell_j}^{u_j} c(p_i, p_j).
\end{aligned}$$

Therefore the cut capacity is a constant, $w(V^-)$, plus the objective function value corresponding to the assigned values of the variables. Thus minimizing the cut is equivalent to minimizing the objective of (BD).

On the other hand, given a finite cut, the solution corresponding to it is feasible: First the source set of the cut contains a consecutive set of nodes $\ell_i, \ldots, p_i$ from each

$x_i$-chain, else the cut would include an infinite capacity arc of the chain and cannot be finite. We let $z_{ij} = \max\{0, p_i - p_j - c_{ij}\}$ which is the number $q$ of nodes in the $x_j$ chain with at least one cut arc adjacent to them. Suppose that $z_{ij}$ is not feasible, i.e., it is greater than $\gamma_{ij}$. Then the infinite capacity arc $(p_i, p_i - c_{ij} - \gamma_{ij})$ is in the cut, contradicting the finiteness of the cut. Therefore the corresponding solution is feasible. □

## 3. A Polynomial Proximity-Scaling Algorithm.

Hochbaum and Shanthikumar [HS] devised a polynomial time algorithm, called the *proximity-scaling algorithm*, for integer convex separable minimization over linear constraints.

A proximity theorem is a statement on the distance, in the $L_\infty$ norm, between the solution to the scaled problem and the optimal solution to the problem. Equivalently, it is a statement on the distance between the optimal solution to the scaled problem with a scaling unit $s$ and the optimal solution to the scaled problem with a scaling unit $s/2$. (Note that 2 can be replaced by any other constant.)

The essence of the proximity-scaling approach is to solve the scaled problem for a scaling unit that is large enough so that the number of binary variables, or pieces, in the piecewise linear approximation is polynomially small. Hochbaum and Shanthikumar [HS] proved a general proximity theorem on the distance between the optimal solutions in the $s$ scaling phase, $\mathbf{x}^s$, and the $s/2$ scaling phase, $\mathbf{x}^{s/2}$:

$$\|\mathbf{x}^s - \mathbf{x}^{\frac{s}{2}}\|_\infty \le n\Delta s,$$

where $\Delta$ is the largest subdeterminant of the constraint matrix and $n$ is the number of variables.[5]

With this proximity theorem, if we choose the scaling unit appropriately, as discussed next, then once the optimal solution in the $s$ scaling phase is available, we know that each variable $x_i^{s/2}$ in the optimal solution for the $s/2$ scaling phase lies in an interval, the size of which is half the size of the previous interval, thus shrinking the range in the next scaling phase by a factor of 2.

For the problem (Convex Dual) the constraint matrix is totally unimodular and thus $\Delta = 1$. We call $\alpha$ the *proximity factor* if $\|\mathbf{x}^s - \mathbf{x}^{\frac{s}{2}}\|_\infty \le \alpha s$. The proximity factor thus proved for (Convex Dual) by Hochbaum and Shanthikumar is $\alpha = n$. In the next section we prove a tighter proximity factor, $\alpha = 1$, applicable to a subclass of (Convex Dual) problems.

Our algorithm is implemented as follows. The scaling unit is selected initially to be $s = U/4\alpha$. The interval for variable $x_j$, $[\ell_j, u_j]$, is thus replaced by up to $4\alpha$ intervals of length $s$ each.

Let the scaled problem called (Dual$^{[s]}$) be defined on the variables $x_j^{[s]} = x_j/s$ and

[5] Although in our problem the number of variables can be $O(m+n)$ we utilize the proximity theorem with regard to the variables $\mathbf{x}$ only. We showed, in a recent working paper [H3], an adaptation of the proximity theorem to *projected* proximity where the proximity is determined by a subset of the variables. Thus in our case we can consider the number of variables to be $n$.

$z_{ij}^{[s]} = z_{ij}/s$:

$$(\mathrm{Dual}^{[s]})\quad \begin{array}{ll} \mathrm{Min} \sum_{j=1}^{n} w_j(s x_j^{[s]}) + \sum_{(i,j)\in E_1} e_{ij}(s z_{ij}^{[s]}) & \\ \text{subject to} \quad x_i^{[s]} - x_j^{[s]} \le \left\lceil \frac{c_{ij}}{s} \right\rceil + z_{ij}^{[s]} & \text{for} \quad (i,j) \in E_1, \\ \qquad x_i^{[s]} - x_j^{[s]} \le \left\lceil \frac{c_{ij}}{s} \right\rceil & \text{for} \quad (i,j) \in E_2, \\ \qquad \frac{u_j}{s} \ge x_j^{[s]} \ge \frac{\ell_j}{s} \ \text{integer} & \text{for} \quad j = 1, \ldots, n, \\ \qquad \frac{\gamma_{ij}}{s} \ge z_{ij}^{[s]} \ge 0 \ \text{integer} & \text{for} \quad (i,j) \in E_1. \end{array}$$

Let the optimal solution to the problem (Dual$^{[s]}$) be $\mathbf{x}^s$. Then according to the proximity theorem the optimal value of $\mathbf{x}^{s/2}$ is within a distance of $\alpha s$ units away, or within an interval of length $2\alpha s$ centered at $\mathbf{x}^s$. This interval, which is guaranteed to contain the optimal solution to $\mathbf{x}^{s/2}$ is then partitioned into at most $4\alpha$ grid points that are spaced at equal distances $s/2$. This procedure then repeats until the scaling unit is $s = 1$.

The pseudopolynomial algorithm described in Section 2 can be applied directly to the problem (Dual$^{[s]}$), solving it by finding a minimum cut on a graph with at most $4\alpha n$ nodes and $4\alpha m$ arcs.

The following is a formal description of the algorithm:

**Proximity-Scaling Algorithm**

```
Step 0. Set $s = \lceil U/4\alpha \rceil$.
Step 1. Solve (Dual$^{[s]}$), with an optimal solution $\mathbf{x}^s$, $\mathbf{z}^s$. If $s = 1$ output the solution and stop.
Step 2. Set $\ell_j \leftarrow \max\{\ell_j, x_j^s - \alpha s\}$ and $u_j \leftarrow \min\{u_j, x_j^s + \alpha s\}$, for $j = 1, \ldots, n$.
Step 3. $s \leftarrow \lceil s/2 \rceil$. Go to step 1.
```

The algorithm calls for (Dual$^{[s]}$) $\log_2 U$ times. The total complexity of solving the problem is thus $\log_2 U \cdot T(4\alpha n, (4\alpha)^2 m)$ when $e_{ij}()$ are convex, or $\log_2 U \cdot T(4\alpha n, 4\alpha m)$ for $e_{ij}()$ linear or piecewise linear with a constant number of pieces.

## 4. A Strong Proximity Theorem.

Our strong proximity theorem applies for the problem on two variables per inequality when the value of $c_{ij}$ is divisible by the scaling unit. This applies, for instance, when all $c_{ij}$ are large enough powers of 2, or 0. Consider the problem (P) with up to two variables per inequality with $f_j()$ convex functions:

$$(\mathrm{P})\quad \begin{array}{ll} \mathrm{Min}\ f(\mathbf{x}) = \sum_{j=1}^{n} f_j(x_j) & \\ \text{subject to} \quad x_i - x_j \ge c_{ij} & \text{for} \quad (i,j) \in E. \end{array}$$

Note that upper and lower bound constraints are included as pairs in $E$ of the form $(i, 0)$, such as $x_i \le u_i$.

We now prove the proximity of the optimal values of the solution for a scaling unit $s$ and for scaling unit $s/2$. This proximity theorem is valid for problems (P) with right-hand sides $c_{ij}$ that are integer multiples of $s$.

*The s-Scaling Phase.* Let $f_j^s(x_j) = f_j(x_j)$ if $x_j$ is an integer multiple of $s$. Let $f_j^s(x_j)$ be defined so that it is linear in its argument between successive integral multiples of $s$. Thus the functions $f_j^s(x_j)$ are piecewise linear approximations of $f_j$. Let (P($s$)) be the problem (P) with $f_j$ replaced by $f_j^s$ for each $j$.

At the $s$-scaling phase, the objective is to find an optimal solution to problem (P($s$)), $\mathbf{x}^s$. Without loss of generality, we restrict attention to optimal solutions for (P($s$)) such that each variable is an integral multiple of $s$.

THEOREM 4.1. *Suppose that* $\mathbf{x}^s$ *is an optimal solution at the* $s$*-scaling phase, and suppose that* $c_{ij}$ *is an integral multiple of* $s$ *for each* $(i, j) \in E$*. Then there is an optimal solution* $\mathbf{x}^*$ *for* $(P(s/2))$ *with the property that* $|x_j^s - x_j^*| \le s$ *for each* $j = 1, \dots, n$.

PROOF. In order to simplify notation, we assume without loss of generality that $s = 2$. We also assume without loss of generality that $\mathbf{x}^s = 0$. (Otherwise, one can perform a translation of variables by replacing $\mathbf{x}$ by $\mathbf{x} - \mathbf{x}^s$.)

Among all optimal solutions to (P) = P($s/2$), let $\mathbf{x}^*$ be one that minimizes

$$\delta = \max_i |x_i^* - x_i^s| = \max_i |x_i^*|.$$

If $\delta \le 2$, there is nothing to prove. So, we assume that $\delta \ge 3$, and we will derive a contradiction. Let $\mathbf{x}'$ be obtained from $\mathbf{x}^*$ as follows:

$$\begin{array}{lll} x_i' = x_i^* + 1 & \text{if} & x_i^* = -\delta, \\ x_i' = x_i^* & \text{if} & -\delta + 1 \le x_i^* \le \delta - 1, \\ x_i' = x_i^* - 1 & \text{if} & x_i^* = \delta. \end{array}$$

Let the objective function value for (P($s$)) be abbreviated as $f^s()$. We claim that $\mathbf{x}'$ is feasible, and that $f^1(\mathbf{x}') \le f^1(\mathbf{x}^*)$. This will contradict that $\mathbf{x}^*$ is an optimal solution to (P(1)) that minimizes $\delta$ and will establish the theorem.

We first show that $\mathbf{x}'$ is feasible. Consider the constraint $x_i - x_j \ge c_{ij}$ for some $(i, j) \in E$. We know that $c_{ij} \le 0$ since the solution $\mathbf{x}^s = 0$ is feasible. Since the constraint is satisfied by $x^*$, the following is true: if $x_i' - x_j' \ge x_i^* - x_j^*$, then $x_i' - x_j' \ge c_{ij}$. So, the only chance for $\mathbf{x}'$ to be infeasible is for there to be some $(i, j) \in E$ such that $x_i' - x_j' < x_i^* - x_j^*$. There are two possible cases in which $x_i' - x_j' < x_i^* - x_j^*$. It is possible that $x_i^* = \delta$ and $x_j^* < \delta$, and it is possible that $x_i^* > -\delta$ and $x_j^* = -\delta$. However, in both of these cases $x_i' - x_j' \ge 0$ and so the constraint $x_i' - x_j' \ge c_{ij}$ is satisfied.

We next show $f^1(\mathbf{x}') \le f^1(\mathbf{x}^*)$. In order to prove this, we create another solution $\mathbf{y}$ as follows:

$$\begin{array}{lll} y_i = -2 & \text{if} & x_i^* = -\delta, \\ y_i = 0 & \text{if} & -\delta + 1 \le x_i^* \le \delta - 1, \\ y_i = 2 & \text{if} & x_i^* = \delta. \end{array}$$

We will show that $\mathbf{y}$ is a feasible solution, and we will also show the following:

$$f^1(\mathbf{x}') - f^1(\mathbf{x}^*) \le \frac{f^2(\mathbf{x}^s) - f^2(\mathbf{y})}{2}.$$

Since $\mathbf{x}^s$ is optimal for (P(2)), it will follow from the inequality above that $(f^2(\mathbf{x}^s) - f^2(\mathbf{y}))/2 \leq 0$, and so $f^1(\mathbf{x}') - f^1(\mathbf{x}^*) \leq 0$, and the theorem will be proved.

We next establish that $\mathbf{y}$ is a feasible solution for (P). Consider a constraint $x_i - x_j \geq c_{ij}$ and recall that $c_{ij} \leq 0$. So if $y_i - y_j \geq 0$, the constraint is satisfied. We consider all three cases in which $y_i - y_j < 0$ as follows:

(i) $-\delta < x_i^* < \delta$ and $x_j^* = \delta$,
(ii) $x_i^* = -\delta$ and $-\delta < x_j^* < \delta$, and
(iii) $x_i^* = -\delta$ and $x_j^* = \delta$.

In cases (i) and (ii), $x_i^* - x_j^* \leq -1$, and so $c_{ij} \leq -1$. Thus these cases cannot happen when $c_{ij} = 0$. Otherwise, since $c_{ij}$ is a multiple of 2, it follows that $c_{ij} \leq -2$, and thus the constraint $y_i - y_j \geq c_{ij}$ is satisfied. In case (iii), $y_i - y_j = -4$ and $x_i^* - x_j^* \leq -6$, and so $y_i - y_j \geq x_i^* - x_j^* \geq c_{ij}$. We thus conclude that $\mathbf{y}$ is feasible.

To complete the proof of the theorem, we need to establish that $f^1(\mathbf{x}') - f^1(\mathbf{x}^*) \leq (f^2(\mathbf{x}^s) - f^2(\mathbf{y}))/2$. We demonstrate that this inequality holds for $i = 1, \ldots, n$.

If $-\delta < x_i^* < \delta$, then $x_i^* = x_i'$ and $y_i = x_i^s$. So, it suffices to focus on indices $i$ for which $x_i^* = -\delta$ or $x_i^* = \delta$. We consider the case that $x_i^* = \delta$. By the convexity of $f_i()$, it follows that $f_i(\delta) - f_i(\delta - 1) \geq (f_i(2) - f_i(0))/2$. Similarly, if $x_i^* = -\delta$, then $f_i(-\delta + 1) - f_i(-\delta) \leq (f_i(0) - f_i(-2))/2$.

Putting these cases together yields that $f^1(\mathbf{x}') - f^1(\mathbf{x}^*) \leq (f^2(\mathbf{x}^s) - f^2(\mathbf{y}))/2$. This completes the proof of the theorem. $\square$

Let all $c_{ij}$ be 0 and those that are positive are $|c_{ij}| = k_{ij}2^{\ell_{ij}}$ for integers $k_{i,j}$. Let $s$ be initially set to $2^\ell$, where $\ell = \min_{i,j} \ell_{ij}$, then the run time of solving for (P(s)) in the first iteration requires $T(n(U/2^\ell), m(U/2^\ell))$ steps. Alternatively, we can use for the first $\log(U/2^\ell)$ steps the standard procedure. The overall run time requires to bring the range of the variable to $2^\ell$ is

$$\min\left\{T\left(n\frac{U}{2^\ell}, m\frac{U}{2^\ell}\right), \log\frac{U}{2^\ell}T(n^2, nm)\right\}.$$

Every one of the subsequent $\ell$ iterations requires $O(T(n, m))$ run time since the $c_{ij}$ are all divisible by the scaling factor. These subsequent iterations are thus accomplished in time $O(\ell T(n, m))$. The total run time is thus

$$O\left(\min\left\{T\left(n\frac{U}{2^\ell}, m\frac{U}{2^\ell}\right), \log\frac{U}{2^\ell}T(n^2, nm)\right\} + \ell T(n, m)\right).$$

Note that the quantity $2^\ell$ can be replaced by any composite number with small factors. The value of $s$ is then updated at each iteration by dividing by one of the factors. The number of grid points then increases by the magnitude of the factor divided by 2.

## 5. A New Algorithm for Minimum Cost Network Flow.

Consider a minimum cost network flow problem defined on a network $G = (V, A)$ with flow variables denoted by $y_{ij}$. Let the sum of supplies be equal to the sum of demands of the nodes, $\sum_{i=1}^{n} w_i = 0$.

Thus we can write the problem's flow balance constraints as inequalities:

$$\begin{aligned}
&\text{Min} \sum_{ij\in A} c_{ij} y_{ij} \\
&\text{subject to} \quad \sum_j y_{ij} - \sum_j y_{ji} \geq w_i, \qquad i \in V, \\
&\qquad\qquad \bar{u}_{ij} \geq y_{ij} \geq 0, \qquad (i,j) \in A.
\end{aligned}$$

The dual of this problem is

$$\begin{aligned}
&\text{Min} \sum_{j\in V} w_j x_j + \sum_{(i,j)\in A} \bar{u}_{ij} z_{ij} \\
&\text{subject to} \quad x_i - x_j \leq c_{ij} + z_{ij} \qquad \text{for} \quad (i,j) \in A, \\
&\qquad\qquad x_j \geq 0, \qquad j = 1, \dots, n, \\
&\qquad\qquad z_{ij} \geq 0, \qquad (i,j) \in A.
\end{aligned}$$

Although there are no explicit upper bounds on the variables $x_j$, it is easy to see that for $C = \max_{(i,j)\in A} c_{ij}$, $x_j \leq nC$ for all $j \in V$. Thus the problem is equivalent to a problem with bounded variables in a range of length $U = nC$. Applying the Proximity-Scaling Algorithm we solve the dual problem as an instance of (Convex Dual) in time $O(T(n^2, mn) \log nC)$.

The solution to the dual problem delivers the node potentials $x_j$ and the dual variables $z_{ij}$. We then construct the solution to the flow problem as follows:

$$y_{ij} = \begin{cases} \bar{u}_{ij} & \text{if} \quad z_{ij} > 0, \\ 0 & \text{if} \quad x_i - x_j < c_{ij}, \\ \in [0, u_{ij}] & \text{if} \quad x_i - x_j = c_{ij}. \end{cases}$$

It remains to balance the supplies and demands in the network while using only arcs in $G$ such that $x_i - x_j = c_{ij}$ (the basic arcs with reduced costs equal to 0). Finding such a feasible flow can be accomplished by solving a maximum flow problem. The complexity of solving the maximum flow problem is dominated by the complexity of solving the dual problem.

To summarize, our algorithm for a minimum cost network flow calls $\log nC$ times for a minimum $s$,$t$-cut procedure, and once for a maximum flow procedure. Therefore the complexity of solving the minimum cost network flow problem using, e.g., Goldberg and Rao's algorithm for the minimum cut procedure, is $O(\min\{n^{4/3}, m^{1/2}n^{1/2}\} mn \log n \log^2(nC))$. Note that there are known maximum flow-based algorithms for the minimum cost network flow problem, by Röck [R] and by Bland and Jensen [BJ]. Both these algorithms have a running time of $O(n \log C \cdot T_{\mathrm{MF}}(n, m))$.

**Acknowledgement.** We thank an anonymous referee for his/her substantial contribution to improving and simplifying the presentation of the three variable per inequality reduction, as well as for numerous other useful editorial suggestions.

## References

[AHO] R. K. Ahuja, D. S. Hochbaum, and J. B. Orlin. Solving the convex cost integer dual network flow problem. *Management Science*, **49**(7), 950–964, 2003.

[AMO] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*: *Theory, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[BBBB] R. E. Barlow, D. J. Bartholomew, J. M. Bremer, and H. D. Brunk. *Statistical Inference Under Order Restrictions*. Wiley, New York, 1972.

[BJ] R. G. Bland and D. L. Jensen. On the computational behavior of a polynomial time network flow algorithm. *Mathematical Programming*, **54**, 1–39, 1992.

[CH] J. Cheriyan and T. Hagerup. A randomized maximum-flow algorithm. *SIAM Journal on Computing*, **24**, 203–226, 1995.

[D] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.

[GR] A. V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, **45**, 783–797, 1998.

[GT1] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. *Journal of the ACM*, **35**, 921–940, 1988.

[GT2] A. V. Goldberg and R. E. Tarjan. Solving minimum cost flow problem by successive approximation. *Mathematics of Operations Research*, **15**, 430–466, 1990.

[H1] D. S. Hochbaum. The pseudoflow algorithm for the maximum flow problem. Manuscript, University of California, Berkeley, CA, 1997.

[H2] D. S. Hochbaum. An efficient algorithm for image segmentation, Markov Random Fields and related problems. *Journal of the ACM*, **48**(4), 686–701, 2001.

[H3] D. S. Hochbaum. The inverse shortest paths problem. Manuscript, University of California, Berkeley, CA, July 2001.

[H4] D. 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.

[H5] D. S. Hochbaum. Efficient algorithms for the inverse spanning tree problem. *Operations Research*, **51**(5), 785–797, 2003.

[HMNT] 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**, 69–83, 1993.

[HN] 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**(6), 1179–1192, 1994.

[HQ] D. S. Hochbaum and M. Queyranne. Minimizing a convex cost closure set. *SIAM Journal of Discrete Mathematics*, **16**(2), 192–207, 2003.

[HS] D. S. Hochbaum and J. G. Shanthikumar. Convex separable optimization is not much harder than linear optimization. *Journal of the ACM*, **37**, 843–862, 1990.

[KM] A. V. Karzanov and S. T. McCormick. Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM Journal on Computing*, **26**(4), 1245–1275, 1997.

[P] J. C. Picard. Maximal closure of a graph and applications to combinatorial problems. *Management Science*, **22**, 1268–1272, 1976.

[R] H. Röck. Scaling techniques for minimum cost network flows. In *Discrete Structures and Algorithms*, V. Pape, ed. Carl Hansen, Munich, pp. 181–191, 1980.