Journal of the Operations Research
Society of Japan
Vol. 35, No. 4, December 1992

DEFERRED RANK ONE UPDATES IN $O(n^3L)$ INTERIOR POINT ALGORITHM

Sanjay Mehrotra
Northwestern University

(Received December 25, 1990; Revised August 27, 1991)

Abstract We present a potential reduction algorithm which performs at most one rank one update at each iteration and at most $O(nL)$ updates to solve a linear program. This algorithm requires $O(n^3L)$ arithmetic operations. An important aspect of this algorithm and its analysis is that it does not always require the approximate scaling matrix to be in a certain box around the primal-dual scaling matrix.

1. Introduction

Anstreicher and Bosch [2], Bosch and Anstreicher [3] showed that the rank one update idea of Karmarkar [6] can be used to improve the worst case complexity of the large step potential reduction algorithm of Ye [14], Freund [4], and Kojima, Mizuno and Yoshise [7]. Let D^k be the scaling matrix in these algorithms. The analysis of Anstreicher and Bosch [2], Bosch and Anstreicher [3], and Karmarkar [6] uses an approximation $\tilde{D}^k$ of D^k as a scaling matrix at every iteration. They assume that at each iteration $\tilde{D}^k$ satisfies

$$1/\rho \leq \tilde{D}^k_{ii}/D^k_{ii} \leq \rho \tag{1.1}$$

for $\rho > 1$, and use

$$\tilde{D}^{k+1}_{ii} = \begin{cases} \tilde{D}^k_{ii}, & \text{if } 1/\rho \leq \tilde{D}^k_{ii}/D^{k+1}_{ii} \leq \rho, \\ D^{k+1}_{ii} & \text{otherwise.} \end{cases} \tag{1.2}$$

This results in a "bulk" rank one updates at each iteration. The Anstreicher and Bosch [2] and Bosch and Anstreicher [3] algorithm terminates in $O(\sqrt{n}L)$ iterations and performs at most $O(\sqrt{n})$ updates on the average. Ye [15] and Mizuno [9] presented $O(n^3L)$ potential reduction algorithm requiring $O(nL)$ iterations.

Mizuno [10] showed that an $O(nL)$ iteration primal-dual algorithm can be developed which performs at most one rank one update at each iteration. The algorithm in Mizuno [10] restricts the ratio of elements of the current point and its approximation so that at most one rank one update is performed at an iteration.

All the rank one update algorithms [2, 5, 6, 9, 10, 11, 13, 15] assume that the approximate scaling matrix satisfy a condition similar to (1.1) at each iteration and update the element not satisfying such a condition. The algorithm in Mizuno [10] restricts the step size, while Anstreicher and Bosch [2] may have to do $O(\sqrt{n})$ updates so that (1.1) is satisfied.

In this paper we show that a strict enforcement of,condition such as (1.1) is not necessary to get $O(nL)$ updates in a $O(n^3L)$ method. Therefore, in the structure of our algorithm we may allow variables to violate (1.1) while taking large steps. The algorithm and its analysis that we present here uses the primal-dual framework. In addition, we differ from Bosch and

Anstreicher [3] in the condition we impose to safeguard the line search, and from Mizuno [10] in the condition we use to check candidate variables for rank one updates. Many other aspects of the analysis in this paper are similar to those in Anstreicher [1], Anstreicher and Bosch[2], Bosch and Anstreicher [3] and Kojima, Mizuno and Yoshise[7].

Notation: X represent a diagonal matrix for which $X_{ii} = x_i$. $\|.\|$ is used to represent two norm. For simplicity $\|u, v\|$ is used to represent two norm of vector $(u^T, v^T)^T$. $P_A = I - A^T(AA^T)^{-1}A$ and $\bar{P}_A = A^T(AA^T)^{-1}A$ orthogonally project a vector onto the ull space and range space of A respectively.

2. Basic Algorithm and its Complexity

Let us consider the linear program (P)

$$\left.\begin{array}{rl} \text{minimize} & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0, \end{array}\right\} \tag{2.1}$$

and its dual (D)

$$\left.\begin{array}{rl} \text{maximize} & b^T \pi \\ \text{s.t.} & A^T\pi + s = c \\ & s \geq 0. \end{array}\right\} \tag{2.2}$$

Consider the primal-dual potential function

$$\phi(x, s) = (n + \sqrt{n}) \ln x^T s - \sum_{i=1}^{n} \ln x_i s_i. \tag{2.3}$$

The derivative of $\phi(x, s)$ is given by

$$\begin{array}{rl} \nabla_x \phi(x, s) & = \frac{n+\sqrt{n}}{x^T s} s - X^{-1} e, \\ \nabla_s \phi(x, s) & = \frac{n+\sqrt{n}}{x^T s} x - S^{-1} e. \end{array} \tag{2.4}$$

In order to solve (P) and (D), for an algorithm it is sufficient to reduce $\phi(x, s)$ by $K\sqrt{n}L$ [4, 14] for some constant K. We now present our algorithm.

Algorithm 2.1:

Input: $A, b, c, x^0, \pi^0, s^0, \tilde{D}^0 = X^{0^{1/2}} S^{0^{-1/2}}, \rho > 1, \beta > 0, \epsilon > 0$.

For $k = 0, 1, \ldots$ do:

$\tilde{D} := \tilde{D}^k$

Obtain the search direction p_x, p_π and p_s by solving the problem $S(\tilde{D})$:

$$\begin{array}{rl} \text{minimize} & -\nabla_x \phi(x, s)^T p_x - \nabla_s \phi(x, s)^T p_s \\ \text{subject to} & A p_x = 0, \\ & A^T p_\pi + p_s = 0, \\ & \tilde{E}(1) \equiv \|\tilde{D}^{-1} p_x, \tilde{D} p_s\| \leq 1. \end{array}$$

The solution of $S(\tilde{D})$ is given by

$$\begin{array}{rl} p_x & = \tilde{D}\hat{p}_x / \|\hat{p}_x, \hat{p}_s\|, \\ p_s & = \tilde{D}^{-1}\hat{p}_s / \|\hat{p}_x, \hat{p}_s\|, \\ p_\pi & = \hat{p}_\pi / \|\hat{p}_x, \hat{p}_s\|, \end{array} \tag{2.5}$$

where

$$\begin{aligned} \hat{p}_x &= P_{A\tilde{D}}\tilde{D}\nabla_x\phi(x,s), \\ \hat{p}_s &= \tilde{P}_{A\tilde{D}}\tilde{D}^{-1}\nabla_s\phi(x,s), \\ \hat{p}_\pi &= -(A\tilde{D}^2A^T)^{-1}A\nabla_s\phi(x,s). \end{aligned}$$

Find a θ by a line search on $\phi(x,s)$ guarded by

$$\phi(x^k,s^k) - \phi(x^k-\theta p_x, s^k-\theta p_s) \geq \frac{\beta}{\sqrt{n}}\sum_{i=1}^{n}\left|\ln\frac{x_i^k/s_i^k}{(x_i^k-\theta(p_x)_i)/(s_i^k-\theta(p_s)_i)}\right| \tag{2.6}$$

and set

$$\begin{aligned} x^{k+1} &= x^k - \theta p_x, \\ \pi^{k+1} &= \pi^k - \theta p_\pi, \\ s^{k+1} &= s^k - \theta p_s. \end{aligned}$$

If $\phi(x^k,s^k) - \phi(x^{k+1},s^{k+1}) \geq \epsilon$, let $\tilde{D}^{k+1} = \tilde{D}^k$,

Otherwise, let $D^{k+1} = X^{k+1^{1/2}}S^{k+1^{-1/2}}$, and

$$\mathcal{V}^k = \{i : D_{ii}^{k+1}/\tilde{D}_{ii}^k > \rho \text{ or } D_{ii}^{k+1}/\tilde{D}_{ii}^k \leq 1/\rho\}. \tag{2.7}$$

If $\mathcal{V}^k \neq \emptyset$ find an $i \in \mathcal{V}^k$ and set $\tilde{D}_{ii}^{k+1} = D_{ii}^{k+1}$.
Also set $\tilde{D}_{ii}^{k+1} = \tilde{D}_{ii}^k$ for all other i.

Obtain $(A(\tilde{D}^{k+1})^2A^T)^{-1}$ from $(A(\tilde{D}^k)^2A^T)^{-1}$ by performing a rank one update.

End

The choice of ρ, ϵ, β used in describing the algorithm is made more precise in Theorem 3.6. In particular, $\rho = 2$, $\beta = .1$ and $\epsilon = .016$ is used in our analysis. The assumption $\tilde{D}^0 = X^{0^{1/2}}S^{0^{-1/2}}$ is for simplicity. We can start with any positive diagonal matrix $\tilde{D}$.

First we explain the logic of Algorithm 2.1. At iteration k we have an approximate scaling matrix $\tilde{D}^k$. A search direction for (P) and (D) is computing from (2.5). We then perform a line search to reduce $\phi(x,s)$ while guarding it by condition (2.6). Condition (2.6) comes out naturally in our analysis. In practice to simplify the line search the right hand side of (2.6) can be replaced by a bound on it (for example use bounds available at any step of the proof of Lemma 3.5). There are two possible cases. **Case I:** A step θ can be taken in the search direction which yield sufficient reduction ϵ in the potential function. In this case we do not update the scaling matrix and proceed to the next iteration.

Case II: The potential function is not reduced by amount ϵ. In this case we find the set of variables which are not in a certain box around the correct scaling matrix. If this set is empty (Case II.1), we proceed to the next iteration as in Case I. Otherwise, we pick any one of the variables in the set, update the corresponding element of the scaling matrix, perform a rank one update to generate projection matrices corresponding to the new scaling matrix, and proceed to the next iteration.

It is important to note that any variable in the set $\mathcal{V}^k$ is a candidate for update. Strategies can be devised to prioritize variables for updating. Furthermore, all the variables in the set $\mathcal{V}^k$ can be updated simultaneously (i.e., "bulk" rank one updates can be performed) without changing the complexity of the algorithm. Therefore, the advantage of a one update at a time approach would be to reduce the total number of rank one updates in practice.

Theorem 2.1. *Assume that at each iteration of Algorithm 2.1 $\phi(x,s)$ is reduced while satisfying (2.6). Then total number of rank one updates and the total number of iterations in this algorithm are bounded by $O(nL)$. Furthermore, this algorithm requires $O(n^3L)$ arithmetic operations to solve a linear program.*

Proof: We first show that it is sufficient to bound the number of rank one updates performed by Algorithm 2.1 by $O(nL)$.

An iteration k could be of two types. If $\tilde{D}^{k+1} = \tilde{D}^k$ we call it Type 1, otherwise we call it Type 2. Let M_1 and M_2 be Type 1 and Type 2 iterations respectively. Let $M = M_1 + M_2$ and let N be the total number of rank one updates.

A Type 1 iteration corresponds to Case I or Case II.1 discussed above. In Case I, $\phi(x,s)$ is reduced by $\epsilon = .016$ at the current iteration. In Case II, from Theorem 3.6, $\phi(x,s)$ is reduced by $\epsilon = .016$ at the next iteration. Since the total desired reduction in $\phi(x,s)$ is $K\sqrt{n}L$, M_1 is bounded by $O(\sqrt{n}L)$. Furthermore, since each Type 2 iteration corresponds to a rank one update, to bound M_2 it is sufficient to bound N. We do this next.

Assume that the scaling matrix corresponding to variable i is updated at iterations k_0^i, k_1^i, $\ldots$, $k_{n_i}^i$. $k_0^i = 0$ for all i. From (2.7) we know that

$$2\ln\rho \le \left|\ln \frac{x_i^{k_l^i}/s_i^{k_l^i}}{x_i^{k_{l+1}^i}/s_i^{k_{l+1}^i}}\right|,$$

then for all updates for variable i we have

$$2n_i \ln\rho \le \sum_{l=0}^{n_i-1}\left|\ln \frac{x_i^{k_l^i}/s_i^{k_l^i}}{x_i^{k_{l+1}^i}/s_i^{k_{l+1}^i}}\right| \le \sum_{l=0}^{M-1}\left|\ln\frac{x_i^l/s_i^l}{x_i^{l+1}/s_i^{l+1}}\right|,$$

and for all variables we have

$$2N\ln\rho \le 2n\ln\rho + \sum_{i=1}^{n}\sum_{l=0}^{M-1}\left|\ln\frac{x_i^l/s_i^l}{x_i^{l+1}/s_i^{l+1}}\right| = n + \sum_{l=0}^{M-1}\sum_{i=1}^{n}\left|\ln\frac{x_i^l/s_i^l}{x_i^{l+1}/s_i^{l+1}}\right|.$$

Now, from (2.6) we have

$$2N\ln\rho \le n + \sqrt{n}/\beta \sum_{l=0}^{M-1}\left(\phi(x^l,s^l) - \phi(x^{l+1},s^{l+1})\right) = O(nL)/\beta.$$

The result follows for appropriate choices of β and ρ. Finally, to prove the complexity of the algorithm it is sufficient to note that the amount of work at Type I and Type II iterations is $O(n^2)$. □

Note that the bound on the number of rank one updates does not require that sufficient progress be made in $\phi(x,s)$ at each iteration. To get this bound it is sufficient that the line searches on $\phi()$ be guarded by (2.6). In the next section we show that if $\mathcal{V}^k = \emptyset$, for $\rho = 2$ and $\beta = .1$, $\phi(x,s)$ can be reduced by $\epsilon = .016$ while satisfying (2.6).

3. Analysis of the Basic Algorithm

First we show that if

$$1/\rho \le \|D^k(\tilde{D}^k)^{-1}\|_\infty \le \rho, \tag{3.1}$$

$\phi(x,s)$ can be reduced sufficiently. Next we will show that it is possible to sufficiently reduce $\phi(x,s)$ while satisfying (2.6). For simplicity we take x^k, π^k, s^k, D^k, $\tilde{D}^k$, x^{k+1},

π^{k+1}, s^{k+1}, D^{k+1}, $\tilde{D}^{k+1}$ as x, π, s, D, $\tilde{D}$, x^+, π^+, s^+, D^+ and $\tilde{D}^+$ respectively. Also let $\delta = \min\{x_i^{1/2}s_i^{1/2}\}$.

Theorem 3.1. *Let $\tilde{D}^k$ satisfy (3.1) and p_x, p_π, p_s be given as in (2.5). Then for some $\tau \in (0,1)$ and θ such that*

$$\theta\|X^{-1}p_x\|_\infty < \tau, \quad \theta\|S^{-1}p_s\|_\infty < \tau, \tag{3.2}$$

we have

$$\phi(x-\theta p_x, s-\theta p_s) \le \phi(x,s) - \frac{\sqrt{3}}{2\rho^2}\tau + \frac{\tau^2}{2(1-\tau)}.$$

We need a few lemmas to prove Theorem 3.1.

Lemma 3.2 *(Kojima, Mizuno and Yoshise [7][page 335]) For a θ satisfying (3.2) we have*

$$\phi(x^+,s^+) - \phi(x,s) \le \theta g_1(p_x,p_s) + \theta^2 g_2(p_x,p_s), \tag{3.3}$$

where

$$g_1(p_x,p_s) = -\frac{n+\sqrt{n}}{x^Ts}(s^Tp_x + x^Tp_s) + e^T(X^{-1}p_x + S^{-1}p_s), \tag{3.4}$$

$$g_2(p_x,p_s) = \|X^{-1}p_x, S^{-1}p_s\|^2/2(1-\tau). \tag{3.5}$$

Lemma 3.2 is standard and we omit its proof here. □

Lemma 3.3 *The directions p_x, p_s obtained from (2.5) satisfy*

$$g_1(p_x,p_s) \le -\frac{\sqrt{3}}{2\rho\delta}.$$

Proof: Let d_x, d_π, d_s solve the problem $S(D)$:

$$\begin{array}{rl} \text{minimize} & -\nabla_x\phi(x,s)^T d_x - \nabla_s\phi(x,s)^T d_s \\ \text{s.t.} & Ad_x = 0 \\ & A^T d_\pi + d_s = 0 \\ & E(1/\rho) \equiv \|D^{-1}d_x, Dd_s\| \le 1/\rho. \end{array}$$

From the definition of $E(1/\rho)$, $\tilde{E}(1)$ and using (3.1) it is easy to see that $E(1/\rho) \subset \tilde{E}(1)$. Therefore d_x, d_π, d_s is a feasible solution for $S(\tilde{D})$. Hence,

$$g_1(p_x,p_s) = -\nabla_x\phi(x,s)^Tp_x - \nabla_s\phi(x,s)^Tp_s \le -\nabla_x\phi(x,s)^Td_x - \nabla_s\phi(x,s)^Td_s. \tag{3.6}$$

The solution of $S(D)$ is given by

$$\begin{array}{ll} d_x & = D\hat{d}_x/(\rho\|\hat{d}_x,\hat{d}_s\|), \\ d_s & = D^{-1}\hat{d}_s/(\rho\|\hat{d}_x,\hat{d}_s\|), \\ d_\pi & = -\hat{d}_\pi/(\rho\|\hat{d}_x,\hat{d}_s\|), \end{array} \tag{3.7}$$

where

$$\begin{array}{ll} \hat{d}_x & = P_{AD}D\nabla_x\phi(x,s), \\ \hat{d}_s & = \bar{P}_{AD}D^{-1}\nabla_s\phi(x,s), \\ \hat{d}_\pi & = (AD^2A^T)^{-1}A\nabla_s\phi(x,s). \end{array}$$

Let

$$d = \frac{n+\sqrt{n}}{x^T y} X^{1/2} S^{1/2} e - X^{-1/2} S^{-1/2} e.$$

Now note that $d = D\nabla_x \phi(x,s) = D^{-1}\nabla_s \phi(x,s)$. Substituting the value of d_x, d_s from (3.7) in (3.6) and using $P_{AD} = P_{AD}P_{AD}$, $\bar{P}_{AD} = \bar{P}_{AD}\bar{P}_{AD}$ and above, we have

$$g_1(p_x, p_s) \le -(d^T P_{AD} d + d^T \bar{P}_{AD} d)/\rho \| P_{AD} d, \bar{P}_{AD} d \| = -\|d\|/\rho.$$

From Kojima, Mizuno and Yoshise [7][Lemma 2.5] we know that $\|d\| \ge \sqrt{3}/2\delta$, which gives the desired result. □

Lemma 3.4.

$$\|X^{-1}p_x\| \le \rho/\delta, \tag{3.8}$$

$$\|S^{-1}p_s\| \le \rho/\delta, \tag{3.9}$$

$$\|X^{-1}p_x,\ S^{-1}p_s\|^2 \le \rho^2/\delta^2. \tag{3.10}$$

Proof:

$$\|X^{-1}p_x\| = \|X^{-1/2}S^{-1/2}D^{-1}\tilde{D}\tilde{D}^{-1}p_x\| \le \|D^{-1}\tilde{D}\tilde{D}^{-1}p_x\|/\delta \le \rho\|\tilde{D}^{-1}p_x\|/\delta \le \rho/\delta.$$

The last two inequalities in above follow from (3.1) and the definition of $\tilde{E}()$ respectively. The proof of (3.9) and (3.10) are similar to the proof of (3.8). □

Proof of Theorem 3.1. From using Lemma 3.3, 3.4 and (3.10) in (3.3) we have

$$\phi(x^+, s^+) - \phi(x^k, s^k) \le -\sqrt{3}\theta/2\rho\delta + \rho^2\theta^2/(2\delta^2(1-\tau)).$$

Let $\theta = \delta\tau/\rho$. For $\tau \in (0,1)$ it is clear from (3.8) and (3.9) that x^+ and s^+ remain feasible. Hence we have

$$\phi(x^+, s^+) - \phi(x^k, s^k) \le -\frac{\sqrt{3}}{2\rho^2}\tau + \frac{\tau^2}{2(1-\tau)}. \square$$

From Theorem 3.1 we know that progress in the potential function can be made for $\rho > 1$. We now show that the condition used to safeguard the line search would also be satisfied for appropriate choices of τ.

Lemma 3.5. *Let p_x, p_π, p_s be given as in (2.5). Then, for $\tau \in (0,1)$ and $\theta = \delta\tau/\rho$ satisfying (3.2) we have*

$$\sum_{i=1}^{n} \left| \ln \frac{x_i/s_i}{x_i^+/s_i^+} \right| \le \sqrt{2n}\tau + \tau^2/2(1-\tau)$$

Proof:

$$\begin{aligned}
\textstyle\sum_{i=1}^n \left|\ln \frac{x_i/s_i}{x_i^+/s_i^+}\right| &\le \textstyle\sum_{i=1}^n (|\ln(1-\theta(X^{-1}p_x)_i)| + |\ln(1-\theta(S^{-1}p_s)_i)|) \\
&\le \theta\|X^{-1}p_x, S^{-1}p_s\|_1 + \theta^2\|X^{-1}p_x, S^{-1}p_s\|^2/2(1-\tau) \\
&\qquad \text{(because } |\ln(1-\theta\nu)| \le \theta|\nu| + \theta^2\nu^2/2(1-\tau),\ \ |\theta\nu| < \tau < 1.) \\
&\le \theta\sqrt{2n}\|X^{-1}p_x, S^{-1}p_s\| + \theta^2\|X^{-1}p_x, S^{-1}p_s\|^2/2(1-\tau) \\
&\le \sqrt{2n}\rho\theta/\delta + \theta^2\rho^2/(2(1-\tau)\delta^2).
\end{aligned}$$

The last inequality follows from using (3.10). Now, by taking $\theta = \delta\tau/\rho$, we have the result. □

Theorem 3.6. *For $\rho = 2$, $\beta = .1$, $\tau = .1$ and $\theta = \delta\tau/\rho$, we have*

$$\phi(x, s) - \phi(x^+, s^+) \geq .016 = \epsilon.$$

Furthermore,

$$\phi(x, s) - \phi(x^+, s^+) \geq \beta/\sqrt{n} \sum_{i=1}^{n} \left| \ln \frac{x_i/s_i}{x_i^+/s_i^+} \right|.$$

Proof. Theorem 3.6 follows by substituting the values of ρ, β and τ in Theorem 3.1 and Lemma 3.5 respectively. □

Acknowledgment: I thank three anonymous referees for their careful reading of this paper and many helpful comments.

References

[1] K.M. Anstreicher (1990). "A standard form variant, and safeguarded line search, for the modified Karmarkar algorithm," *Mathematical Programming,* 47, 337-351.

[2] K.M. Anstreicher and R.A. Bosch (1988). " Long steps in a $O(n^3L)$ algorithm for linear programming," Technical Report, Yale School of Management, New Haven, CT.

[3] R.A. Bosch and K.M. Anstreicher (1991). "On partial updating in a potential reduction linear programming algorithm algorithm of Kojima, Mizuno and Yoshise," Technical Report, Yale School of Management, New Haven, CT. (To appear in *Algorithmica*).

[4] R.M. Freund (1988). "Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function," OR182-88, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA.

[5] C.C. Gonzaga (1988). "An algorithm for solving linear programs in $O(n^3L)$ operations," in N. Megiddo, ed., *Progress in Mathematical Programming, Interior Point and Related Methods,* Springer-Verlag, New York, pp. 1-28.

[6] N. Karmarkar (1984). "A new polynomial-time algorithm for linear programming," *Combinatorica,* 4, 373-395.

[7] M. Kojima, S. Mizuno and A. Yoshise (1991). "An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems," *Mathematical Programming,* 50(3), 331-342.

[8] S. Mizuno (1991). "$O(n^aL)$ iteration and $O(n^3L)$-Operation potential reduction algorithms for linear programming," *Linear Algebra and Its Applications*, 152, 155-168.

[9] S. Mizuno (1990). "An $O(n^3L)$ algorithm using a sequence for a linear complementarity problem," *Journal of the Operations Research Society of Japan,* 33, No. 1, 66-75.

[10] S. Mizuno (1990). "A Rank One Updating Interior Algorithm for Linear Programming," *Arabian Journal for Science and Engineering,* 15 (4B), 671-677.

[11] R. C. Monteiro and I. Adler (1989). "Interior path following primal-dual algorithms. part I: Linear programming," *Mathematical Programming,* 44, 27-42.

[12] M. J. Todd and Y. Ye (1990). "A centered projective algorithm for linear programming," *Mathematics of Operations Research,* 15(3), 508-529.

[13] P. M. Vaidya (1990). "An algorithm for linear programming which requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations," *Mathematical Programming,* 47(2), 175-201.

[14] Y. Ye (1991). "An $O(n^3L)$ potential reduction algorithm for linear programming," *Mathematical Programming,* 50(2), 239-258.

[15] Y. Ye (1989). "Further developments in potential reduction algorithm," Technical Report, Department of Management Sciences, The University of Iowa, Iowa.

Sanjay Mehrotra
Department of Industrial Engineering
and Management Sciences
Northwestern University
Evanston, IL 60208
USA