A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems: A Summary

Masakazu Kojima* Nimrod Megiddo† Toshihito Noma‡ Akiko Yoshise§

Revised October 1990

Abstract. This note summarizes a report with the same title, where a study was carried out regarding a unified approach, proposed by Kojima, Mizuno and Yoshise, for interior point algorithms for the linear complementarily problem with a positive semi-definite matrix. This approach is extended to nonsymmetric matrices with nonnegative principal minors.

Keywords. interior-point methods, complementarity, linear programming, quadratic programming, path-following, potential reduction

1. Introduction.

Many algorithms (see [10] for references) have been developed for mathematical programming since Karmarkar's paper [8]. See [22] for a survey. We consider here the linear complementarity problem (LCP): Given $\boldsymbol{M} \in R^{n\times n}$ and $\boldsymbol{q} \in R^n$, find $(\boldsymbol{x}, \boldsymbol{y}) \in R^{2n}$ such that

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q},\ (\boldsymbol{x}, \boldsymbol{y}) \geq \boldsymbol{0} \ \text{ and } \ x_i y_i = 0\ (i \in N = \{1, \ldots, n\}). \tag{1.1}$$

Kojima, Mizuno and Yoshise [14] proposed an $O(\sqrt{n}L)$ iteration potential reduction algorithm. They suggested a unified interior point (UIP) method for both the path-following and potential reduction algorithms that solve LCP's with positive semi-definite (p.s.d.) matrices. The purpose of this note is to summarize a study which the authors have carried out in [10] about the UIP method.

*Departments of Information Sciences and Systems Science, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, Japan.

†IBM Almaden Research Center, San Jose, CA 95120, and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel.

‡Department of Systems Science, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, Japan.

§Graduate School of Systems Management, The University of Tsukuba, Otsuka, Bunkyo-ku, Tokyo 112, Japan.

The general LCP is NP-complete [3]. In [10] we assume $\boldsymbol{M}$ to be in the rich class P_0 of matrices with nonnegative principal minors. This class includes not only skew-symmetric matrices, p.s.d. matrices and P-matrices, but also NP-complete LCP's.

It is very insightful to consider the space (primal, dual, or primal-dual) in which various interior point algorithms work. Consider the pair of quadratic programs:

$$\text{P} : \text{Minimize } \boldsymbol{c}^T\boldsymbol{u} + \frac{1}{2}\boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{u} \text{ subject to } \boldsymbol{A}\boldsymbol{u} \geq \boldsymbol{b},\ \boldsymbol{u} \geq \boldsymbol{0}.$$

$$\text{D} : \text{Maximize } \boldsymbol{b}^T\boldsymbol{v} - \frac{1}{2}\boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{u} \text{ subject to } \boldsymbol{A}^T\boldsymbol{v} - \boldsymbol{Q}\boldsymbol{u} \leq \boldsymbol{c},\ \boldsymbol{v} \geq \boldsymbol{0},$$

where $\boldsymbol{Q}$ is symmetric and p.s.d. If $\boldsymbol{Q} = \boldsymbol{O}$, we get a symmetric primal-dual pair of linear programs. We can state the P and D together as an LCP (1.1) with the p.s.d. matrix by defining

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{Q} & -\boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{O} \end{pmatrix}, \quad \boldsymbol{q} = \begin{pmatrix} \boldsymbol{c} \\ -\boldsymbol{b} \end{pmatrix}.$$

The UIP method serves as a primal-dual interior point algorithm if we focus our attention to LCPs arising from linear programming and convex quadratic programming. In fact, the UIP method is closely related to or includes as special cases many interior point algorithms (for example, [12; 16; 17], etc.) which work on the primal and dual spaces simultaneously. The global and the polynomial-time convergence results which are established in [10] can be applied to a wider class of primal-dual interior point algorithms. The UIP method can also be applied to linear programming problems in standard form (see [14]).

Consider first a smooth version of the method. We define a vector field, induced by the Newton directions towards the central trajectory, on the set S_{++} of all the interior feasible solutions of the LCP. The vector field over S_{++} defines a system of differential equations such that each solution forms a trajectory (smooth curve) through each point $(\boldsymbol{x}^0, \boldsymbol{y}^0) \in S_{++}$ toward a solution of the LCP. Furthermore, the Newton directions are descent directions of the potential function whose value tends to $-\infty$ as a point $(\boldsymbol{x}, \boldsymbol{y})$ approaches a solution of the LCP along any trajectory.

The potential reduction algorithm [14] can be obtained from the smooth version by specifying the step size at each $(\boldsymbol{x}, \boldsymbol{y}) \in S_{++}$ for numerical integration of the vector field, using the first order method for differential equations. If in addition we choose an initial point sufficiently close to the trajectory and a suitably small step size, the potential reduction algorithm works as the path-following algorithm [13]. For more details see Section 4.3 of [10].

Our work is closely related to the works [19; 20; 21] by Tanabe (see [10] for a detailed discussion), but the analyses are different and, in particular, we emphasize the global and the polynomial-time convergence of the UIP method applied to larger classes of LCP's.

We do not give a unified view over all the interior point algorithms. Specifically, we do not cover algorithms which work only on either primal or dual space [5; 8; 18], the $O(\sqrt{n}L)$ iteration potential reduction algorithms of [25; 7], the first potential reduction algorithm given in [11] for P_0 case. We summarize the main results of [10]. In Section 2, we state some classes of LCP's to which we apply the UIP method. Section 3 explains the UIP method. Section 4 gives some assumptions which are necessary for the discussion of the theoretical computational complexity of the UIP method. Section 5 presents a large class of potential reduction algorithms as special cases of the UIP method, and their global and polynomial-time convergence properties.

2. LCPs Covered by the UIP Method.

Let $R^n_+ = \{\boldsymbol{x} \in R^n : \boldsymbol{x} \geq \boldsymbol{0}\}$ and $R^n_{++} = \{\boldsymbol{x} \in R^n : \boldsymbol{x} > \boldsymbol{0}\}$. A pair $(\boldsymbol{x}, \boldsymbol{y})$ is called a *feasible solution* if

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q},\ (\boldsymbol{x}, \boldsymbol{y}) \geq \boldsymbol{0},$$

and an *interior* feasible solution if

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q},\ (\boldsymbol{x}, \boldsymbol{y}) > \boldsymbol{0}.$$

We denote

$$\begin{aligned} S_+ &= \{(\boldsymbol{x}, \boldsymbol{y}) \in R^{2n}_+ : \boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}\}, \\ S_{++} &= \{(\boldsymbol{x}, \boldsymbol{y}) \in R^{2n}_{++} : \boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}\}, \\ S_{cp} &= \{(\boldsymbol{x}, \boldsymbol{y}) \in S_+ : x_i y_i = 0\ (i \in N)\}. \end{aligned}$$

It is convenient to consider the quadratic programming problem

$$\text{QP} : \text{Minimize}\ \ \boldsymbol{x}^T\boldsymbol{y}\ \ \text{subject to}\ \ (\boldsymbol{x}, \boldsymbol{y}) \in S_+. \tag{2.2}$$

The LCP is equivalent to the QP in the sense that $(\boldsymbol{x}, \boldsymbol{y})$ is a solution of the LCP if and only if it is a minimum solution of the QP with the objective value zero. We will describe the UIP method in Section 3, and then derive in Section 5 a class of potential reduction algorithms as special cases. Under the condition stated below, each algorithm in the class converges globally, i.e., it generates a bounded sequence $\{(\boldsymbol{x}^k, \boldsymbol{y}^k)\} \subset S_{++}$ such that $\lim_{k\to\infty} \boldsymbol{x}^{kT}\boldsymbol{y}^k = 0$; hence the sequence has at least one accumulation point, which is a solution of the LCP.

Condition 2.1. We will assume that the matrix M satisfies the following conditions:

(i) The matrix $\boldsymbol{M}$ is a P_0-matrix, i.e., has nonnegative principal minors.

(ii) A point $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ which lies in the interior S_{++} of the feasible region S_+ is known.
(iii) The set $S_+^t = \{(\boldsymbol{x}, \boldsymbol{y}) \in S_+ : \boldsymbol{x}^T\boldsymbol{y} \leq t\}$ is bounded for every $t \geq 0$.

We need (i) to guarantee the existence and uniqueness of a solution to a system of linear equations for the search directions. The class P_0 contains the following classes:

SS : skew-symmetric matrices $\boldsymbol{M}$, characterized by $\boldsymbol{\xi}^T\boldsymbol{M}\boldsymbol{\xi} = 0$ for every $\boldsymbol{\xi} \in R^n$.
PSD : positive semi-definite matrices.
P : matrices with positive principal minors.
$P_*(\kappa)$: matrices $\boldsymbol{M}$ satisfying $(1+4\kappa)\sum_{i\in I_+(\xi)} \xi_i[\boldsymbol{M}\boldsymbol{\xi}]_i + \sum_{i\in I_-(\xi)} \xi_i[\boldsymbol{M}\boldsymbol{\xi}]_i \geq 0$ for every $\boldsymbol{\xi} \in R^n$, where $I_+(\boldsymbol{\xi}) = \{i \in N : \xi_i[\boldsymbol{M}\boldsymbol{\xi}]_i > 0\}$ and $I_-(\boldsymbol{\xi}) = \{i \in N : \xi_i[\boldsymbol{M}\boldsymbol{\xi}]_i < 0\}$, and $\kappa \geq 0$.
P_* : $\bigcup_{k\geq 0} P_*(\kappa)$.
CS : "column sufficient" matrices [4], characterized by the condition that $\xi_i[\boldsymbol{M}\boldsymbol{\xi}]_i \leq 0$ $(i \in N)$ always implies $\xi_i[\boldsymbol{M}\boldsymbol{\xi}]_i$ $(i \in N)$.

Note that $SS \subset PSD$, $P \cap SS = \emptyset$, and $(PSD \cup P) \subset P_* \subset CS \subset P_0$.

Requirement (ii) offers an initial point $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ while (iii) ensures the boundedness of the generated sequence $\{(\boldsymbol{x}^k, \boldsymbol{y}^k)\} \subset S_{++}$. Requirements (ii) and (iii) are not generally satisfied. If $\boldsymbol{M}$ is column sufficient, the original problem (LCP) can be transformed into an "artificial" problem (LCP') of order $2n$ such that (a) (LCP') satisfies Condition 2.1 and (b) we either obtain a solution of (LCP) by deleting the artificial components from the solution of (LCP') or conclude that (LCP) has no solution. Thus, the UIP method solves the LCP if $\boldsymbol{M} \in CS$.

3. The UIP Method.

One of the main ingredients of the UIP method is the function

$$f(\boldsymbol{x}, \boldsymbol{y}) = (n+\nu)\log \boldsymbol{x}^T\boldsymbol{y} - \sum_{i=1}^{n} \log x_i y_i - n \log n \ . \tag{3.3}$$

Here $\nu > 0$ is a parameter. This type of potential function has been utilized for linear programs in many papers (see [10]). Associated with the quadratic programming problem into which we converted the LCP we consider the potential minimization problem:

$$\text{Minimize } \ f(\boldsymbol{x}, \boldsymbol{y}) \ \text{ subject to } \ (\boldsymbol{x}, \boldsymbol{y}) \in S_{++}.$$

Note that the term $(n+\nu)\log \boldsymbol{x}^T\boldsymbol{y}$ of f comes from the objective function, whereas $-\sum_{i=1}^{n} \log x_i y_i$ works as a "barrier." We rewrite the definition of f as:

$$f(\boldsymbol{x}, \boldsymbol{y}) \ = \ \nu f_{cp}(\boldsymbol{x}, \boldsymbol{y}) + f_{cen}(\boldsymbol{x}, \boldsymbol{y}), \tag{3.4}$$

$$
\begin{aligned}
f_{cp}(\boldsymbol{x},\boldsymbol{y}) &= \log \boldsymbol{x}^T\boldsymbol{y}, \\
f_{cen}(\boldsymbol{x},\boldsymbol{y}) &= n\log \boldsymbol{x}^T\boldsymbol{y} - \sum_{i=1}^{n}\log x_i y_i - n\log n = n\log \frac{\boldsymbol{x}^T\boldsymbol{y}/n}{(\prod_{i=1}^{n} x_i y_i)^{1/n}} .
\end{aligned}
$$

The term $\dfrac{\boldsymbol{x}^T\boldsymbol{y}/n}{(\prod_{i=1}^{n} x_i y_i)^{1/n}}$ is the ratio of the arithmetic mean and the geometric mean of n positive numbers $x_1y_1, x_2y_2, \ldots, x_ny_n$. Hence $f_{cen}(\boldsymbol{x},\boldsymbol{y}) \geq 0$ for every $(\boldsymbol{x},\boldsymbol{y}) \in S_{++}$. It follows that $f(\boldsymbol{x},\boldsymbol{y}) \geq \nu f_{cp}(\boldsymbol{x},\boldsymbol{y}) = \nu \log \boldsymbol{x}^T\boldsymbol{y}$ for every $(\boldsymbol{x},\boldsymbol{y}) \in S_{++}$. Thus we may regard $(\boldsymbol{x},\boldsymbol{y}) \in S_{++}$ as an approximate solution if $f(\boldsymbol{x},\boldsymbol{y})$ is sufficiently small.

The function f assists in choosing the step lengths. It also helps establishing either global or polynomial-time convergence. It is worth noting that the quantity $\boldsymbol{x}^T\boldsymbol{y}$ may approach zero even when $f(\boldsymbol{x},\boldsymbol{y})$ is bounded from below, so the design of algorithms should not rely solely on the analysis of f.

The other main ingredient of the UIP method is the central trajectory for the LCP [15; 13], given by

$$
S_{cen} = \{(\boldsymbol{x},\boldsymbol{y}) \in S_{++} : \boldsymbol{X}\boldsymbol{y} = t\boldsymbol{e} \quad \text{for some } t > 0\}, \tag{3.5}
$$

where $\boldsymbol{e} = (1,\ldots,1)^T \in R^n$ and $\boldsymbol{X} = \operatorname{diag}\boldsymbol{x}$.

Under certain assumptions, the central trajectory S_{cen} indeed constitutes a smooth curve which leads to a solution of the LCP. The formula for the central trajectory S_{cen} can be rewritten using the function f_{cen}:

$$
S_{cen} = \{(\boldsymbol{x},\boldsymbol{y}) \in S_{++} : f_{cen}(\boldsymbol{x},\boldsymbol{y}) = 0\}. \tag{3.6}
$$

The search direction is chosen as follows. Let $0 \leq \beta \leq 1$. Given $(\boldsymbol{x},\boldsymbol{y}) \in S_{++}$, assign $(d\boldsymbol{x}, d\boldsymbol{y}) \in R^{2n}$ such that

$$
\begin{pmatrix} \boldsymbol{Y} & \boldsymbol{X} \\ -\boldsymbol{M} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} d\boldsymbol{x} \\ d\boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \beta\dfrac{\boldsymbol{x}^T\boldsymbol{y}}{n}\boldsymbol{e} - \boldsymbol{X}\boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix}. \tag{3.7}
$$

Here $\boldsymbol{Y} = \operatorname{diag}\boldsymbol{y}$. The coefficient matrix on the left-hand side above is nonsingular whenever $\boldsymbol{M} \in P_0$. This matrix, as well as the right-hand side vector, are smooth in $(\boldsymbol{x},\boldsymbol{y}) \in S_{++}$, so the $(d\boldsymbol{x}, d\boldsymbol{y})$'s form a smooth vector field over S_{++}.

The vector $(d\boldsymbol{x}, d\boldsymbol{y})$ is the Newton direction at $(\boldsymbol{x},\boldsymbol{y})$ for approximating a point $(\boldsymbol{x}',\boldsymbol{y}')$ on the central trajectory such that

$$
\boldsymbol{X}'\boldsymbol{y}' = \beta\frac{\boldsymbol{x}^T\boldsymbol{y}}{n}\boldsymbol{e} \quad \text{and} \quad \boldsymbol{y}' = \boldsymbol{M}\boldsymbol{x}' + \boldsymbol{q}, \tag{3.8}
$$

where $\boldsymbol{X}' = \operatorname{diag}\boldsymbol{x}'$. The system (3.8) involves the parameter $\beta \in [0,1]$. Let the point $(\boldsymbol{x}',\boldsymbol{y}')$ on S_{cen} denote the solution of the system of equations (3.8). If $\beta = 1$, the

point $(\boldsymbol{x}', \boldsymbol{y}')$ corresponds to the point $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ which minimizes the Euclidean distance $\|\boldsymbol{X}\boldsymbol{y} - \hat{\boldsymbol{X}}\hat{\boldsymbol{y}}\|$ from the current point $(\boldsymbol{x}, \boldsymbol{y})$ to a point $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ on the central trajectory. Thus the direction $(d\boldsymbol{x}, d\boldsymbol{y}) = (d\boldsymbol{x}^c, d\boldsymbol{y}^c)$ may be regarded as a "centering direction" in this case. On the other hand, if $\beta = 0$, the system (3.8) together with the nonnegativity condition on $(\boldsymbol{x}', \boldsymbol{y}')$ coincides with the LCP itself. Hence $(d\boldsymbol{x}, d\boldsymbol{y})$ gives the Newton direction to the LCP without any centering. We will call $(d\boldsymbol{x}, d\boldsymbol{y}) = (d\boldsymbol{x}^a, d\boldsymbol{y}^a)$ an "affine scaling direction." In general, $(d\boldsymbol{x}, d\boldsymbol{y})$ can be represented as a convex combination of the centering direction $(d\boldsymbol{x}^c, d\boldsymbol{y}^c)$ and the affine scaling direction $(d\boldsymbol{x}^a, d\boldsymbol{y}^a)$.

$$(d\boldsymbol{x}, d\boldsymbol{y}) = \beta(d\boldsymbol{x}^c, d\boldsymbol{y}^c) + (1 - \beta)(d\boldsymbol{x}^a, d\boldsymbol{y}^a). \tag{3.9}$$

For $\beta \in [0, 1]$, it can be shown that $(d\boldsymbol{x}, d\boldsymbol{y})$ is a descent direction of f.

Henceforth we assume an initial point $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in S_{++}$ is available, and the initial value of the potential function is not greater than a certain constant.

The UIP Method.

Step 1: Let $k = 1$ and $\epsilon > 0$.
Step 2: Let $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^k, \boldsymbol{y}^k)$. Stop if the inequality $\boldsymbol{x}^T\boldsymbol{y} \leq \epsilon$ is satisfied.
Step 3: Let $\beta = \beta_k \in [0, 1]$. Solve the system (3.7) for a search direction $(d\boldsymbol{x}, d\boldsymbol{y})$.
Step 4: Choose $\theta = \theta_k \geq 0$ so that $(\boldsymbol{x}, \boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y}) \in S_{++} \cup S_{cp}$. Let the new point be $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) = (\boldsymbol{x}, \boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y})$.
Step 5: Let $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}) = (\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$. Replace k by $k + 1$, and go to Step 2.

The stopping criterion in Step 2 covers the case where we have reached a solution of the LCP. In practice, one can expect to obtain a sufficiently approximate solution of the LCP when the method stops, provided $\epsilon > 0$ is small enough. In theory we choose $\epsilon = 2^{-2L}$ so that a solution of the LCP can be computed from the terminal point of the UIP method (see [10]).

The UIP method involves two parameters: $\beta \in [0, 1]$ and $\theta \geq 0$. The parameter β and the current point $(\boldsymbol{x}^k, \boldsymbol{y}^k)$ determine the direction $(d\boldsymbol{x}, d\boldsymbol{y})$ toward the new iterate $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1})$, while $\theta \geq 0$ controls the step length. In [10] we explore flexible choices of these two parameters which ensure the global convergence, in certain cases in polynomial time. A summary is presented in Section 5.

4. Assumptions for Evaluating the Computational Complexity.

We assume all the entries of $\boldsymbol{M}$ and $\boldsymbol{q}$ to be rational. This is necessary to guarantee that the LCP in the cases under consideration, will have a solution with rational coordinates

which could be computed in a finite number of arithmetic operations. For simplicity, we further assume all the entries to be integral. The *size* L of the LCP is defined by:

$$L = \sum_{i=1}^{n}\sum_{j=1}^{n} \lceil \log_2(|m_{ij}| + 1) \rceil + \sum_{i=1}^{n} \lceil \log_2(|q_i| + 1) \rceil + 2 \lceil \log_2(n+1) \rceil + n(n+1),$$

($\boldsymbol{M} = (m_{ij})$). It is shown in [13] that if an approximate solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \in S_+$ satisfies $\hat{\boldsymbol{x}}^T \hat{\boldsymbol{y}} \leq 2^{-2L}$, then there exists a solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ such that $x_i^* = 0$ if $\hat{x}_i \leq 2^{-L}$ and $y_j^* = 0$ if $\hat{y}_j \leq 2^{-L}$. Such a $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ can be computed in $O(n^3)$ arithmetic operations. Under Condition 2.1 we can stop when ${\boldsymbol{x}^k}^T \boldsymbol{y}^k \leq 2^{-2L}$ and get a solution. In other words, it is theoretically sufficient to choose $\epsilon = 2^{-2L}$. Practically, however, it might be too complicated to compute with such a small number. We often assume:

Condition 4.1.

(i) The entries of $\boldsymbol{M}$ and $\boldsymbol{q}$ are integral.
(ii) The matrix $\boldsymbol{M}$ belongs to $P_*(\kappa)$ for some $\kappa \geq 0$.
(iii) A point $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in S_{++}$ with $f_{cp}(\boldsymbol{x}^1, \boldsymbol{y}^1) = O(L)$ and $f_{cen}(\boldsymbol{x}^1, \boldsymbol{y}^1) \leq \alpha$ is known, where $\alpha > 0$ is given.

Condition 4.1 implies Condition 2.1. Hence the potential reduction algorithm of Section 5 solves the problem under Condition 4.1. In [10] we present a method of transforming a given LCP satisfying only (i) and (ii) of Condition 4.1 into an equivalent artificial problem satisfying (i), (ii) and (iii) simultaneously.

5. Globally Convergent Potential Reduction Algorithms.

A path-following algorithm traces the central trajectory numerically by generating a sequence $\{(\boldsymbol{x}^k, \boldsymbol{y}^k)\}$ in "a horn neighborhood" of S_{cen} such that $\lim_{k\to\infty} {\boldsymbol{x}^k}^T \boldsymbol{y}^k = 0$. Several ways to define a neighborhood of S_{cen} are presented in [10]. Many path-following algorithms are special cases of the UIP method [6; 12; 13; 16; 17; 23]. Specifically, if we define the neighborhood by $\|\boldsymbol{X}\boldsymbol{y} - (\boldsymbol{x}^T\boldsymbol{y})/n)\boldsymbol{e}\| \leq \alpha(\boldsymbol{x}^T\boldsymbol{y})/n$ with $\alpha \in (0, 0.2]$, $\beta_k = 1 - \delta/\sqrt{n}$, $(\delta = \alpha/(1-\alpha))$ and $\theta_k = 1$, we get the $O(n^{3.5}L)$ path-following algorithm for the p.s.d. case of [13]. A narrow neighborhood of S_{cen} yields polynomial-time algorithms but is not good for practical purposes. Let $\alpha > 0$ and consider a neighborhood $N_{cen}(\alpha)$ defined by $f_{cen}(\boldsymbol{x}, \boldsymbol{y}) \leq \alpha$ as in [19; 20; 21]. When $\alpha = \infty$, we define $N_{cen}(\infty) = S_{++}$. Recall that S_{cen} corresponds to $f_{cen}(\boldsymbol{x}, \boldsymbol{y}) = 0$. Hence S_{cen} coincides with $N_{cen}(0)$. Since f_{cen} is continuous, for each $\alpha > 0$, the subset $(\boldsymbol{x}, \boldsymbol{y}) \in S_{++}$ such that $f_{cen}(\boldsymbol{x}, \boldsymbol{y}) < \alpha$ is open relative to S_{++}.

The potential reduction algorithm described below may be viewed as a path-following algorithm in the sense that all its iterates lie in a certain neighborhood $N_{cen}(\alpha_{bd})$ of S_{cen} with $0 < \alpha_{bd} < \infty$ or $\alpha_{bd} = \infty$ (the case of ∞ will discussed later). Its main features are: (i) a wider neighborhood $N_{cen}(\alpha_{bd})$ of S_{cen} which contains any given point in S_{++} (specifically, when $N_{cen}(\infty) = S_{++}$ it works as a potential reduction algorithm), and (ii) each iteration reduces the value of f but not necessarily that of $\boldsymbol{x}^T\boldsymbol{y}$. When a narrow neighborhood of S_{cen} is taken, the algorithm is path-following. However, with a wider neighborhood of S_{cen}, it seems natural to regard it as a modification of a potential reduction algorithm. This suggests flexibility in designing practically efficient algorithms.

Given an arbitrary $\alpha_{bd} > 0$ or $= \infty$ and an initial $(\boldsymbol{x}^1, \boldsymbol{y}^1) \in N_{cen}(\alpha_{bd})$, we want to choose β and θ so that the generated sequence $\{(\boldsymbol{x}^k, \boldsymbol{y}^k)\}$ stays in $N_{cen}(\alpha_{bd})$ and $\lim_{k\to\infty} \boldsymbol{x}^{kT}\boldsymbol{y}^k = 0$. Thus, we need to move toward the center by taking a larger $\beta \leq 1$ if we are close to the boundary of $N_{cen}(\alpha_{bd})$. On the other hand, if we are far from the boundary, or if we are close to the S_{cen}, we can take a smaller $\beta \geq 0$ to get more reduction in $\boldsymbol{x}^T\boldsymbol{y}$. To embody this idea, we introduce parameters α_{cen}, α_1, β_{cen} and β_{bd} such that

$$\left.\begin{array}{l} 0 < \alpha_{cen} \leq \alpha_1 < \alpha_{bd} < \infty \quad \text{or} \\ 0 < \alpha_{cen} \leq \alpha_1 \leq \alpha_{bd} = \infty, \\ 0 \leq \beta_{cen} < 1, \quad 0 < \beta_{bd} \leq 1, \end{array}\right\} \tag{5.10}$$

and then choose the direction parameter β at each iteration such that

$$\left.\begin{array}{ll} 0 \leq \beta \leq \beta_{cen} & \text{if } f_{cen}(\boldsymbol{x}, \boldsymbol{y}) < \alpha_{cen}, \\ 0 \leq \beta \leq 1 & \text{if } \alpha_{cen} \leq f_{cen}(\boldsymbol{x}, \boldsymbol{y}) \leq \alpha_1, \\ \beta_{bd} \leq \beta \leq 1 & \text{if } \alpha_1 < f_{cen}(\boldsymbol{x}, \boldsymbol{y}). \end{array}\right\} \tag{5.11}$$

The idea of taking a larger $\beta \leq 1$ when the current point is far from the central trajectory was proposed in [2] (See also [26; 27; 14].).

The vector $(d\boldsymbol{x}, d\boldsymbol{y})$ depends on the value of β. The function $f = \nu f_{cp} + f_{cen}$ facilitates the choice of $\theta = \theta_k$ for the step length. Since $(\boldsymbol{x}, \boldsymbol{y})$ lies in S_{++} and $(d\boldsymbol{x}, d\boldsymbol{y})$ satisfies the Newton equation (3.7), $\boldsymbol{y} + \theta d\boldsymbol{y} = \boldsymbol{M}(\boldsymbol{x} + \theta d\boldsymbol{x}) + \boldsymbol{q}$ for every $\theta \geq 0$. Let $\theta^* > 0$ be the minimizer of $f((\boldsymbol{x}, \boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y}))$ subject to $(\boldsymbol{x}, \boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y}) \in N_{cen}(\alpha_{bd}) \cup S_{cp}$ or, equivalently, either $f_{cen}((\boldsymbol{x}, \boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y})) \leq \alpha_{bd}$ or $(\boldsymbol{x}, \boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y}) \in S_{cp}$. It is generally impossible to compute θ^*. Let $0 \leq \tau < 1$, and

$$\Theta(\tau) = \sup\{\theta \geq 0 : \theta d\boldsymbol{x} \geq -\tau\boldsymbol{x}, \quad \theta d\boldsymbol{y} \geq -\tau\boldsymbol{y}\}.$$

Obviously, for every $\theta \in [0, \Theta(\tau)]$, $(\boldsymbol{x}, \boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y}) \in S_{++}$. It is possible (see Section 4.4 of [10]) to find quadratic functions $G^\tau_{cen}(\theta)$ and $G^\tau(\theta)$ such that

$$f_{cen}((\boldsymbol{x}, \boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y})) \leq f_{cen}(\boldsymbol{x}, \boldsymbol{y}) - G^\tau_{cen}(\theta) \quad \text{for every } \theta \in [0, \Theta(\tau)],$$

$$G^\tau_{cen}(0) = 0, \quad \frac{dG^\tau_{cen}(0)}{d\theta} > 0,$$
$$f((\boldsymbol{x},\boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y})) \le f(\boldsymbol{x},\boldsymbol{y}) - G^\tau(\theta) \quad \text{for every } \theta \in [0, \Theta(\tau)],$$
$$G^\tau(0) = 0, \quad \frac{dG^\tau(0)}{d\theta} > 0.$$

Thus, using the minimizer, $\hat{\theta}$, of $f(\boldsymbol{x},\boldsymbol{y}) - G^\tau(\theta)$ subject to

$$f_{cen}(\boldsymbol{x},\boldsymbol{y}) - G^\tau_{cen}(\theta) \le \alpha_{bd} \quad \text{and} \quad \theta \in [0, \Theta(\tau)],$$

we obtain a new point $(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}) = (\boldsymbol{x},\boldsymbol{y}) + \hat{\theta}(d\boldsymbol{x}, d\boldsymbol{y})$ such that

$$(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}) \in N_{cen}(\alpha_{bd}),$$
$$f((\boldsymbol{x},\boldsymbol{y}) + \theta^*(d\boldsymbol{x}, d\boldsymbol{y})) \le f(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}) \le f(\boldsymbol{x},\boldsymbol{y}) - G^\tau(\hat{\theta}) < f(\boldsymbol{x},\boldsymbol{y}).$$

Theoretically, $\hat{\theta}$ ensures either the global or the polynomial-time convergence with suitable choices of β. Practically, it gives an initial value for an inexact line search for a better approximation of θ^*.

To summarize the convergence results so far, let α_{cen}, α_1, α_{bd}, β_{cen} and β_{bd} satisfy (5.10). Let $(\boldsymbol{x}^1,\boldsymbol{y}^1) \in N_{cen}(\alpha_{bd})$. Also, assume β is chosen so that (5.11) holds, and that $\theta \ge 0$ ensures

$$(\boldsymbol{x},\boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y}) \in N_{cen}(\alpha_{bd}),$$
$$f((\boldsymbol{x},\boldsymbol{y}) + \theta(d\boldsymbol{x}, d\boldsymbol{y})) \le f((\boldsymbol{x},\boldsymbol{y}) + \hat{\theta}(d\boldsymbol{x}, d\boldsymbol{y})).$$

We will specify β_{cen} and β_{bd} to ensure the global convergence, in polynomial-time in some cases.

First, suppose that Condition 2.1 is satisfied. Then the sequence $\{(\boldsymbol{x}^k,\boldsymbol{y}^k)\}$ is bounded and $\lim_{k\to\infty} {\boldsymbol{x}^k}^T \boldsymbol{y}^k = 0$ (See [10].). Three important cases are covered by this result:

(a) $0 < \alpha_{cen} \le \alpha_1 < \alpha_{bd} < \infty$.
(b) $0 < \alpha_{cen} \le \alpha_1 < \alpha_{bd} = \infty$.
(c) $0 < \alpha_{cen} \le \alpha_1 = \alpha_{bd} = \infty$.

The second result is an extension of the polynomial-time convergence result of [14] for the p.s.d. case. Suppose the LCP satisfies Condition 4.1. Let $\nu = \sqrt{n}$, $\alpha_1 < +\infty$ and $\beta_{cen} = \beta_{bd} = n/(n+\nu)$. Then the potential reduction algorithm solves the LCP in $O(\sqrt{n}(1+\kappa)L)$ iterations. We may take either a finite $\alpha_{bd} > \alpha_1$ as in the case (a) or $\alpha_{bd} = +\infty$ as in the case (b). See [10].

The last result we state here is a special case of the case (c) of the first convergence result above. Suppose the LCP satisfies Condition 4.1. Let $\nu = \sqrt{n}$. Choose $\alpha_{cen} =$

$\alpha_1 = \alpha_{bd} = \infty$ and $\beta_{cen} = 0$. This choice implies $\beta_k = 0$ throughout. The algorithm then solves the LCP in $O(\exp\{\sqrt{n}(1+\kappa)L\})$ iterations (See [10].). This potential reduction algorithm with $\beta_k = 0$ ($k = 1, 2, \ldots$) is a direct application of the (damped) Newton method to the system of equations

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}, \quad x_i y_i = 0 \ (i \in N)$$

associated with the LCP. The algorithm may also be regarded as an affine scaling interior point algorithm for the LCP because the search direction at each iteration involves no centering force as in the affine scaling interior point algorithms for linear programs ([1; 5; 24], etc.). We acknowledge a conversation with I. Lustig on this subject. The global convergence of the algorithm presented here does not rely on nondegeneracy assumptions, while the global convergence of the affine scaling algorithms for linear programs has been established under such assumptions.

More general global convergence theorems are established in Section 6.2 of [10] from which we can directly derive the results above.

Acknowledgements. Most of this research was done while three of the authors, Kojima, Noma and Yoshise were visiting at the IBM Almaden Research Center in the summer of 1989. Partial support from the Office of Naval Research under Contract N00014-87-C-0820 is acknowledged. Masakazu Kojima also acknowledges partial support from the Grant-in-Aids for Co-operative Research (63490010) of The Ministry of Education, Science and Culture of Japan.

References

[1] E. R. Barnes. A variation on Karmarkar's algorithm for solving linear programming problems. *Math. Programming*, 36:174–182, 1986.

[2] E. R. Barnes, S. Chopra and D. L. Jensen. A polynomial time version of the affine algorithm. Tech. Rep., IBM T. J. Watson Res. Ctr., Yorktown Heights, N.Y., 1988.

[3] S. J. Chung. A note on the complexity of the LCP: the LCP is strongly NP-complete. Tech. Rep. 792, Dept. of Industrial and Operations Engineering, Univ. of Michigan, Ann Arbor, MI, 1979.

[4] R. W. Cottle, J.-S. Pang and V. Venkateswaran. Sufficient matrices and the linear complementarity problem. *Linear Algebra and Its Applications*, 114/115:231–249, 1989.

[5] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Math. Dokl.*, 8:674–675, 1967.

[6] J. Ding and T.-Y. Li. A polynomial-time predictor-corrector algorithm for linear complementarity problems. Tech. Rep., Dept. of Math., Michigan State Univ., East Lansing, MI, 1989.

[7] R. M. Freund. Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. OR 182-88, Sloan School of Manag., M.I.T., Cambridge, MA, 1988.

[8] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[9] M. Kojima, N. Megiddo and T. Noma. Homotopy continuation methods for nonlinear complementarity problems. *Math. of Oper. Res.*, to appear.

[10] M. Kojima, N. Megiddo, T. Noma and A. Yoshise. *A Unified approach to interior point algorithms for linear complementarity problems.* Lecture Notes in Computer Science, Springer-Verlag, to appear. Also issued as Res. Rep. RJ 7493, IBM Almaden Res. Ctr., San Jose, CA, 1990.

[11] M. Kojima, N. Megiddo and Y. Ye. An interior point potential reduction algorithm for the linear complementarity problem. *Math. Programming*, to appear.

[12] M. Kojima, S. Mizuno and A. Yoshise. A primal-dual interior point algorithm for linear programming. In N. Megiddo, ed., *Progress in Math. Programming, Interior-Point and Related Methods*, pp. 29–47, New York, 1989. Springer-Verlag.

[13] M. Kojima, S. Mizuno and A. Yoshise. A polynomial-time algorithm for a class of linear complementary problems. *Math. Programming*, 44:1–26, 1989.

[14] M. Kojima, S. Mizuno and A. Yoshise. An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems. *Math. Programming*, to appear.

[15] N. Megiddo. Pathways to the optimal set in linear programming. In N. Megiddo, ed., *Progress in Math. Programming, Interior-Point and Related Methods*, pp. 131–158, New York, 1989. Springer-Verlag.

[16] R. D. C. Monteiro and I. Adler. Interior path following primal-dual algorithms. Part I: Linear programming. *Math. Programming*, 44:27–41, 1989.

[17] R. D. C. Monteiro and I. Adler. Interior path following primal-dual algorithms. Part II: Convex quadratic programming. *Math. Programming*, 44:43–66, 1989.

[18] J. Renegar. A polynomial-time algorithm based on Newton's method for linear programming. *Math. Programming*, 40:59–94, 1988.

[19] K. Tanabe. Center flattening transformation and a centered Newton method for linear programming. Tech. Rep., The Inst. of Stat. Math., Minato-ku, Tokyo, Japan, 1987.

[20] K. Tanabe. Complementarity-enforcing centered Newton method for mathematical programming. In K. Tone, ed., *New Methods for Linear Programming*, pp. 118–144, The Inst. of Stat. Math., Minato-ku, Tokyo, Japan, 1987.

[21] K. Tanabe. Centered Newton method for mathematical programming. In M. Iri and K. Yajima, eds., *System Modelling and Optimization*, pp. 197–206. Springer-Verlag, 1988.

[22] M. J. Todd. Recent developments and new directions in linear programming. Tech. Rep. No. 827, School of O.R. and I.E., Cornell Univ., Ithaca, NY, 1988.

[23] M. J. Todd and Y. Ye. A centered projective algorithm for linear programming. Tech. Rep. No. 763, School of O.R. and I.E., Cornell Univ., Ithaca, NY, 1987.

[24] R. J. Vanderbei, M. S. Meketon and B. A. Freedman. A modification of Karmarkar's linear programming algorithm. *Algorithmica*, 1:395–407, 1986.

[25] Y. Ye. A class of potential functions for linear programming. Tech. Rep., Dept. of Engineering-Economic Sys., Stanford Univ., CA, 1988.

[26] Y. Ye. Line search in potential reduction algorithms for linear programming. Tech. Rep., Dept. of Manag. Sci., Univ. of Iowa, Iowa City, IA, 1989.

[27] Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. Tech. Rep., Dept. of Manag. Sci., Univ. Iowa, Iowa City, IA, 1989.