Algorithmica (1986) 1: 455–482

Algorithmica
© 1986 Springer-Verlag New York Inc.

A Multiplicative Barrier Function Method for Linear Programming[1]

Masao Iri[2] and Hiroshi Imai[3]

Abstract. A simple Newton-like descent algorithm for linear programming is proposed together with results of preliminary computational experiments on small- and medium-size problems. The proposed algorithm gives local superlinear convergence to the optimum and, experimentally, shows global linear convergence. It is similar to Karmarkar's algorithm in that it is an interior feasible direction method and self-correcting, while it is quite different from Karmarkar's in that it gives superlinear convergence and that no artificial extra constraint is introduced nor is projective geometry needed, but only affine geometry suffices.

Key Words. Linear programming, Interior method, Barrier function, Newton method.

1. Introduction.

Since Khachian's epoch-making work [5], [6],[4] we have gradually been recognizing that the "theoretical" property of an iterative LP-algorithm being of polynomial order is realized if there is an appropriately defined auxiliary function, such as the volume and/or the thickness of the ellipsoid in Khachian's case [5], [6], and the potential function in Karmarkar's case [4], which is reduced at each iteration step (which is to be performed in polynomial time) by a constant factor, and if the ratio of the largest possible value of the function over the smallest possible value (the latter is a kind of threshold of resolution proper to the problem in correspondence to the precision of input data) is not larger than a polynomial-order power of that constant.

However, at the same time we have known that such a linear convergence alone, especially if the reduction rate is small, is by no means of practical use. A "practically efficient" (in the natural sense of the words) algorithm, if it is of the iterative type, should have a large reduction rate, and might preferably show superlinear convergence at least at the final stages. We also know that

[1] The works of the first author and the second were supported in part by the Grant-in-Aid for Scientific Research (B) 60460130 (1985) and by the Grant-in-Aid for Encouragement of Young Scientists (A) 60790046 (1985), respectively, of the Ministry of Education, Science and Culture of Japan.
[2] Department of Mathematical Engineering and Instrumentation Physics, Faculty of Engineering, University of Tokyo, Bunkyo-ku, Tokyo, Japan 113.
[3] Department of Computer Science and Communication Engineering, Kyushu University, Hakozaki, Fukuoka, Japan 812.
[4] With regard to Khachian's papers [5], [6], the latter is a substantial improvement on the former so that we should refer to the latter when we discuss Khachian's work although the former is still often referred to in Western literature. Moreover, it is regretful that the effect of rounding errors due to finite-precision computation as well as Khachian's ingenious device for circumventing it, is not always emphasized.

Received September 20, 1985; revised July 16, 1986. Communicated by Nimrod Megiddo.

the theoretical complexity of an algorithm and its practical efficiency are not very well correlated with each other, especially when the complexity is large. (In contrast, when the time complexity is linear or quadratic in the problem size, it is more likely that they are better correlated.) Proof of the polynomiality of an algorithm is one thing, the design of a practically efficient one is another; both being of independent importance.

The general idea of the algorithm proposed in this paper was conceived by the first author with the proof of the convexity of the objective function as well as of the superlinearity of convergence, whereas the second author refined the design and analysis of the algorithm, implemented the idea into a computer program for computational-experimental use, and carried out rather extensive systematic computational experiments.

Part I (Sections 2-6) of this note was written mainly by the first author while Part II (Section 7-12) was written by the second.

This paper is an improved version of the note which was presented by the first author at the 12th International Symposium on Mathematical Programming [2]. (It was, substantially, an extended English version of the hand-out [3], written in Japanese, which was distributed at the meeting of the Research Group on Mathematical Programming of the Operations Research Society of Japan held on February 16, 1985.) The authors owe much of the improvement to personal suggestions from many participants of that symposium, among whom they mention with gratitude Professor O. L. Mangasarian of the University of Wisconsin and Dr. R. J. Vanderbei of AT&T Bell Laboratories. They thank the anonymous referees whose comments were useful in improving not only the presentation but also the essential contents of the paper.

Part I. Theory

2. Problem. The problem we consider in the following is to minimize the objective function

(2.1)[5]
$$c(x) \equiv \sum_{\kappa=1}^{n} c_\kappa x^\kappa - c_0$$

under the inequality constraints

(2.2)
$$a^i(x) \equiv \sum_{\kappa=1}^{n} a^i_\kappa x^\kappa - a^i_0 \geq 0 \qquad (i=1,\ldots,m),$$

where c_0, c_κ, a^i_0, and a^i_κ ($\kappa=1,\ldots,n$; $i=1,\ldots,m$) are given constants.

[5] Almost everywhere throughout the paper, we shall adopt the tensor notation for indices ($\kappa, \lambda, \ldots$) of vectors and tensors in $\mathbf{R}^n$ because we believe that invariant characters of the relevant quantities and equations can best be visualized by doing so. At least the distinction between the superscripts standing for contravariance and the subscripts for covariance must be noticed. However, we shall not use Einstein's summation convention (if we did so, we would write, e.g., $c_\kappa x^\kappa$ for $\sum_{\kappa=1}^{n} c_\kappa x^\kappa$) for fear that, although tensorial symbolism is one of the most fundamental pieces of knowledge in classical mathematics, many readers may not be very familiar with it.

Following Karmarkar [4] we assume without loss in generality that the interior Int X of the feasible region

$$X \equiv \{x \in \mathbf{R}^n \mid a^i(x) \geq 0 \ (i=1, \ldots, m)\} \tag{2.3}$$

is nonempty and a strictly interior point $x^{(0)} \in \operatorname{Int} X$:

$$a^i(x^{(0)}) > 0 \qquad (i=1, \ldots, m) \tag{2.4}$$

is given, and that an optimum solution exists and the optimum (i.e., the minimum) value of the objective function is *a priori* known to be equal to zero:

$$c(\hat{x}) = \min\{c(x) \mid x \in X\} = 0. \tag{2.5}$$

Note that (2.5) implies that $c(x) \geq 0$ at every point x in the feasible region X.

Furthermore, we exclude some trivial cases from our consideration by adopting assumptions (i)-(iii), where we denote the set of optimum solutions by $\hat{X}$:

$$\hat{X} = \{x \in X \mid c(x) = 0\}. \tag{2.6}$$

(i) $\hat{X} \neq X$, i.e., $c(x) > 0$ in Int X.
(ii) $\hat{X}$ is bounded.
(iii) At a basic optimum solution (the existence of which is assured by assumption (ii)) there is at least one inactive constraint.

Condition (i) can be easily checked.

Condition (ii) is assumed here so that the sequence produced by the proposed algorithm does not diverge to infinity. As long as the sequence converges to a point of $\hat{X}$, condition (ii) is not necessary. If a tendency for the sequence to diverge is detected, we may add an extra constraint so that the set of optimum solutions becomes bounded. In fact, as is well known [6], if there is an optimum solution at all, there is one such that the values of the components do not exceed a bound determined readily from the input data.

Condition (iii) is satisfied, for example, if the feasible region is bounded. The case in which (iii) fails to be satisfied can be handled trivially (see Proposition 3.5).

3. A Multiplicative Barrier Function and Its Derivatives.

We define a new function $F(x)$ made up of the objective function $c(x)$ and the constraint functions $a^i(x)$, which will play the central role in our algorithm, as

$$F(x) \equiv c(x)^{m+1} \Big/ \prod_{i=1}^{m} a^i(x), \tag{3.1}$$

which is defined only in the interior Int X of the feasible region X.

Under the assumptions we took in Section 2, it is readily seen that

$$F(x) > 0 \qquad \text{in Int } X. \tag{3.2}$$

Apparently, this is the affine analogue of Karmarkar's potential function [4], but it has a number of nice properties (the convexity property in particular) which can be easily observed as follows.

PROPOSITION 3.1. *If $F(x^{(\nu)}) \to 0$ for a sequence of interior feasible points $x^{(\nu)} \in \operatorname{Int} X$ $(\nu = 0, 1, 2, \ldots)$, then the distance between $x^{(\nu)}$ and $\hat{X}$ converges to* 0 (*and hence, if there is a unique optimum, the sequence converges to it*).

PROOF. If the set of points $\{x^{(\nu)}\}$ is bounded, so are the $a^i(x^{(\nu)})$'s. Therefore $F(x^{(\nu)}) \to 0$ implies $c(x^{(\nu)}) \to 0$, and, due to assumption (i), the distance between $x^{(\nu)}$ and $\hat{X}$ tends to 0. If the set of points $\{x^{(\nu)}\}$ is not bounded, it might be possible that, because $a^i(x^{(\nu)})$ became large, $c(x^{(\nu)})$ did not converge to 0 even if $F(x^{(\nu)}) \to 0$. However, even in such a case, there would be a constant b^i (>0) for each i such that $a^i(x)/c(x) \le b^i$, since there is no infinite feasible ray parallel to $\hat{X}$ due to assumption (ii), which would lead us to a contradiction that $F(x^{(\nu)}) = c(x^{(\nu)})^{m+1}/\prod_{i=1}^m a^i(x^{(\nu)}) \ge b \cdot c(x^{(\nu)})$ (where $b = \prod_{i=1}^m b^i$) did not converge to 0. □

We might have chosen, instead of $m+1$, a number greater than the number of active constraints at the optimum point in order to have only Propositon 3.1, but it will be seen that the choice of "$m+1$" or larger is also essential to the strict convexity of $F(x)$ as will be seen in (3.9) and the proof to Propositon 3.3. (cf. also Section 6.3).

The converse of Proposition 3.1 does not hold in general, but we have the following propositon instead.

PROPOSITION 3.2. *If the sequence $\{x^{(\nu)}\}$ converges to the optimum in a certain closed polyhedron P such that $\hat{X} \subset P \subset \hat{X} \cup \operatorname{Int} X$* (*including as a special case the convergence along a straight line*), *then $F(x^{(\nu)})$ tends to 0 as $\nu \to \infty$.*

PROOF. Since P is a closed polyhedron and $\operatorname{Int} X$ is an open polyhedron, for any $x \in P$ with $c(x) = \varepsilon$ (>0), there is a constant b^i (>0) such that $a^i(x) > b^i \cdot c(x)$ for each i. Hence $F(x) < c(x) \cdot \prod_{i=1}^m b^i$. □

Thus, in order to find an optimum solution to our linear programming problem we may find a sequence of points $x^{(0)}, x^{(1)}, \ldots$ in $\operatorname{Int} X$ such that the sequence $F(x^{(0)}), F(x^{(1)}), \ldots$ rapidly converges to zero.

It is interesting to see that the derivatives of $F(x)$ have nice expressions, as follows. To begin with we differentiate the logarithm of $F(x)$:

$$\log F(x) = (m+1) \log c(x) - \sum_{i=1}^{m} \log a^i(x) \tag{3.3}$$

to get

$$\eta_\kappa(x) \equiv \frac{\partial}{\partial x^\kappa} \log F(x) = \frac{1}{F(x)} \frac{\partial}{\partial x^\kappa} F(x) = (m+1)\frac{c_\kappa}{c(x)} - \sum_{i=1}^m \frac{a_\kappa^i}{a^i(x)}. \tag{3.4}$$

The vector $\boldsymbol{\eta} = \eta_\kappa$, which is actually the gradient of $F(x)$ divided by $F(x)$, will simply be called the "gradient" in the following. Denote

$$\begin{aligned} \tilde{c}_\kappa(x) &\equiv \frac{c_\kappa}{c(x)}, \\ \tilde{a}_\kappa^i(x) &\equiv \frac{a_\kappa^i}{a^i(x)}, \\ \bar{a}_\kappa(x) &\equiv \frac{1}{m}\sum_{i=1}^m \tilde{a}_\kappa^i(x). \end{aligned} \tag{3.5}$$

Then we can write

$$\eta_\kappa(x) = (m+1)\tilde{c}_\kappa(x) - m\bar{a}_\kappa(x). \tag{3.6}$$

Further differentiation will yield

$$\begin{aligned} B_{\lambda\kappa}(x) &\equiv \frac{\partial^2}{\partial x^\lambda \partial x^\kappa} \log F(x) = \frac{1}{F(x)} \frac{\partial^2}{\partial x^\lambda \partial x^\kappa} F(x) - \eta_\lambda(x)\eta_\kappa(x) \\ &= -(m+1)\tilde{c}_\lambda(x)\tilde{c}_\kappa(x) + \sum_{i=1}^m \tilde{a}_\lambda^i(x)\tilde{a}_\kappa^i(x). \end{aligned} \tag{3.7}$$

Thus, the Hessian matrix of $F(x)$ divided by $F(x)$, which we shall simply call "the Hessian" of $F(x)$ in the following, is

$$H_{\lambda\kappa}(x) \equiv \frac{1}{F(x)} \frac{\partial^2}{\partial x^\lambda \partial x^\kappa} F(x) = B_{\lambda\kappa}(x) + \eta_\lambda(x)\eta_\kappa(x). \tag{3.8}$$

It is an amusing exercise to rewrite the expression for $H_{\lambda\kappa}$ using (3.6) and (3.7) by completing squares as follows:

$$\begin{aligned} H_{\lambda\kappa}(x) &= m^2[\tilde{c}_\lambda(x) - \bar{a}_\lambda(x)][\tilde{c}_\kappa(x) - \bar{a}_\kappa(x)] \\ &\quad + \sum_{i=1}^m [\tilde{c}_\lambda(x) - \tilde{a}_\lambda^i(x)][\tilde{c}_\kappa(x) - \tilde{a}_\kappa^i(x)] \\ &= m(m+1)[\tilde{c}_\lambda(x) - \bar{a}_\lambda(x)][\tilde{c}_\kappa(x) - \bar{a}_\kappa(x)] \\ &\quad + \sum_{i=1}^m [\tilde{a}_\lambda^i(x) - \bar{a}_\lambda(x)][\tilde{a}_\kappa^i(x) - \bar{a}_\kappa(x)]. \end{aligned} \tag{3.9}$$

Expression (3.9) itself shows that $H_{\lambda\kappa}$ is nonnegative definite, so that $F(x)$ is convex in Int X. Furthermore, under assumption (iii) of Section 2, the positive definiteness of $H_{\lambda\kappa}$ can be proved as follows, where we note the well-known fact that the nonnegative (resp. positive) definiteness of the Hessian over an open domain implies the convexity (resp. strict convexity) over the domain.

PROPOSITION 3.3. *$H_{\lambda\kappa}$ is positive definite so that $F(x)$ is strictly convex in* Int X.

PROOF. We have already seen that $H_{\lambda\kappa}$ is nonnegative definite. If $H_{\lambda\kappa}$ is not positive definite at a point $x \in \text{Int}\, X$, there is a nonvanishing vector ξ such that

$$\begin{aligned}\sum_{\lambda=1}^{n}\sum_{\kappa=1}^{n} H_{\lambda\kappa}\xi^{\kappa}\xi^{\lambda} = m(m+1)\left\{\sum_{\kappa=1}^{n}[\tilde{c}_{\kappa}(x)\xi^{\kappa} - \bar{a}_{\kappa}(x)\xi^{\kappa}]\right\}^2 \\ + \sum_{i=1}^{m}\left\{\sum_{\kappa=1}^{n}[\tilde{a}^{i}_{\kappa}(x)\xi^{\kappa} - \bar{a}_{\kappa}(x)\xi^{\kappa}\right\}^2 = 0,\end{aligned} \tag{3.10}$$

which means

$$\sum_{\kappa=1}^{n}\tilde{c}_{\kappa}(x)\xi^{\kappa} = \sum_{\kappa=1}^{n}\tilde{a}^{i}_{\kappa}(x)\xi^{\kappa} \equiv g \qquad (i=1,\dots,m). \tag{3.11}$$

From the set $I=\{i \mid a^{i}(\hat{x})=0\}$ of active constraints at a basic optimum solution $\hat{x}$ we can then choose $\hat{I}$ such that $\{a^{i}_{\kappa} \mid i\in\hat{I}\}$ is a maximal independent subset (i.e., a basis) of $\{a^{i}_{\kappa} \mid i \in I\}$. Then, since

$$g = \sum_{\kappa=1}^{n}\tilde{a}^{i}_{\kappa}(x)\xi^{\kappa} = \frac{1}{a^{i}(x)}\sum_{\kappa=1}^{n}a^{i}_{\kappa}\xi^{\kappa} = 0 \qquad \text{for} \quad i\in\hat{I} \tag{3.12}$$

would imply $\xi^{\kappa}=0$, g cannot vanish. Furthermore, for any linear form

$$b(x) = \sum_{\kappa=1}^{n} b_{\kappa}x^{\kappa} - b_0, \tag{3.13}$$

there is a set of coefficients β_i such that

$$b_{\kappa} = \sum_{i\in\hat{I}}\beta_i a^{i}_{\kappa} \tag{3.14}$$

and

$$b(x)-b(\hat{x}) = \sum_{i\in\hat{I}}\beta_i[a^{i}(x)-a^{i}(\hat{x})] = \sum_{i\in\hat{I}}\beta_i a^{i}(x). \tag{3.15}$$

If there is an inactive constraint $a^{i_0}(\hat{x}) \neq 0$, then, taking $a^{i_0}(x)$ for $b(x)$, we have

$$\begin{aligned} g a^{i_0}(x) &\neq g[a^{i_0}(x) - a^{i_0}(\hat{x})] \\ &= g\sum_{i\in\hat{I}}\beta_i a^{i}(x) = \sum_{i\in\hat{I}}\sum_{\kappa=1}^{n}\beta_i a^{i}_{\kappa}\xi^{\kappa} \\ &= \sum_{\kappa=1}^{n} a^{i_0}_{\kappa}\xi^{\kappa} = g a^{i_0}(x),\end{aligned} \tag{3.16}$$

which is a contradiction. □

PROPOSITION 3.4. *The direction* $\boldsymbol{\xi}$ *determined by*

$$\sum_{\kappa=1}^{n} H_{\lambda\kappa}(x)\xi^{\kappa} = -\eta_{\lambda}(x) \tag{3.17}$$

is a strict descent direction of $F(x)$ *at* $x \in \operatorname{Int} X$.

PROOF. We have $\sum_{\lambda=1}^{n} \eta_{\lambda}\xi^{\lambda} = -\sum_{\lambda=1}^{n}\sum_{\kappa=1}^{n} H_{\lambda\kappa}(x)\xi^{\kappa}\xi^{\lambda}$, so that we have from the strict convexity of $H_{\lambda\kappa}$

$$\sum_{\kappa=1}^{n} \eta_{\kappa}\xi^{\kappa} < 0. \tag{3.18}$$ □

PROPOSITION 3.5. If assumption (iii) does not hold, $F(x)$ is linear along the rays emanating from a unique optimum solution $\hat{x}$.

PROOF. There is a unique optimum solution since, otherwise, assumption (ii) would imply assumption (iii). For any feasible x, we have

$$\begin{aligned} a^{i}(x) &= a^{i}(x) - a^{i}(\hat{x}) = \sum_{\kappa=1}^{n} a^{i}_{\kappa} \cdot (x^{\kappa} - \hat{x}^{\kappa}), \\ c(x) &= c(x) - c(\hat{x}) = \sum_{\kappa=1}^{n} c_{\kappa} \cdot (x^{\kappa} - \hat{x}^{\kappa}). \end{aligned} \tag{3.19}$$

Therefore, on the line $x(t) = \hat{x} + t\boldsymbol{\xi}$ with a constant vector $\boldsymbol{\xi}$, we have

$$F(x(t)) = t \cdot \left(\sum_{\kappa=1}^{n} c_{\kappa}\xi^{\kappa}\right)^{m+1} \Bigg/ \prod_{i=1}^{m} \left(\sum_{\kappa=1}^{n} a^{i}_{\kappa}\, \xi^{\kappa}\right), \tag{3.20}$$

i.e., the function $F(x)$ is linear along the rays emanating from $\hat{x}$. □

4. Algorithm.

The algorithm we propose is straightforward on the basis of the observation we made in the previous section.

1. Start from the given initial point $x^{(0)} \in \operatorname{Int} X$.
2. Iteration:

   At the νth approximation $x^{(\nu)}$, compute $F^{(\nu)} = F(x^{(\nu)})$ (from the viewpoint of numerical computation we should not compute $F^{(\nu)}$ itself but $\log F^{(\nu)}$), $\eta^{(\nu)}_{\kappa} = \eta_{\kappa}(x^{(\nu)})$ and $H^{(\nu)}_{\lambda\kappa} = H_{\lambda\kappa}(x^{(\nu)})$ by (3.3), (3.4), (3.7), and (3.8) (or (3.9)), and then solve the system of linear equations

$$\sum_{\kappa=1}^{n} H^{(\nu)}_{\lambda\kappa}\xi^{(\nu)\kappa} = -\eta^{(\nu)}_{\lambda} \qquad (\lambda = 1, \ldots, n) \tag{4.1}$$

   to determine the vector $\xi^{(\nu)\kappa}$.

Perform the line search in the direction of $\boldsymbol{\xi}^{(\nu)}$ to find the minimum of $F(\boldsymbol{x})$ on that line, i.e., determine t^* by

$$\left[\frac{d}{dt}F(\boldsymbol{x}^{(\nu)}+t\boldsymbol{\xi}^{(\nu)})\right]_{t=t^*}=0 \tag{4.2}$$

and set

$$\boldsymbol{x}^{(\nu+1)}=\boldsymbol{x}^{(\nu)}+t^*\boldsymbol{\xi}^{(\nu)}. \tag{4.3}$$

In the algorithm proposed above, it is theoretically assumed that we can obtain the exact value of t^*. Computationally, the line search in each iteration step can be performed very quickly, and, from the point of view of the polynomiality of the algorithm, even the bisection search would be sufficient to determine t^* to within the "threshold of resolution" [6]. Practically, we may adopt any reasonable method and a more hasty stopping rule therefore. Here, we have

$$\begin{gathered}
f(t)=F(\boldsymbol{x}^{(\nu)}+t\boldsymbol{\xi}^{(\nu)})=\frac{[c(\boldsymbol{x}^{(\nu)})+\gamma t]^{m+1}}{\prod_{i=1}^{m}[a^i(\boldsymbol{x}^{(\nu)})+\alpha^i t]},\\
\gamma=\sum_{\kappa=1}^{n}c_\kappa\xi^{(\nu)\kappa},\quad \alpha^i=\sum_{\kappa=1}^{n}a^i_\kappa\xi^{(\nu)\kappa}\quad (i=1,\ldots,m),\\
\frac{1}{f(t)}\frac{d}{dt}f(t)=(m+1)\frac{\gamma}{c(\boldsymbol{x}^{(\nu)})+\gamma t}-\sum_{i=1}^{m}\frac{\alpha^i}{a^i(\boldsymbol{x}^{(\nu)})+\alpha^i t},\\
\frac{1}{f(t)}\frac{d^2}{dt^2}f(t)=\left[\frac{1}{f(t)}\frac{d}{dt}f(t)\right]^2-(m+1)\left[\frac{\gamma}{c(\boldsymbol{x}^{(\nu)})+\gamma t}\right]^2+\sum_{i=1}^{m}\left[\frac{\alpha^i}{a^i(\boldsymbol{x}^{(\nu)})+\alpha^i t}\right]^2.
\end{gathered} \tag{4.4}$$

Note here that $[df/dt]_{t=0}=-F(\boldsymbol{x}^{(\nu)})\sum_{\lambda=1}^{n}\sum_{\kappa=1}^{n}H^{(\nu)}_{\lambda\kappa}\xi^{(\nu)\kappa}\xi^{(\nu)\lambda}<0$.

Since $f(t)$ itself is strictly convex and tends to infinity near the boundary $t=\bar{t}$ where

$$\bar{t}=\min_i\{-a^i(\boldsymbol{x}^{(\nu)})/\alpha^i \mid \alpha^i<0\}, \tag{4.5}$$

we may first search for the smallest t_1 for which $df/dt>0$ among the sequence $((1-\frac{1}{2})\bar{t}, (1-\frac{1}{4})\bar{t}, (1-\frac{1}{8})\bar{t},\ldots)$ and then, starting from t_1, apply the Newton iteration

$$t_{j+1}=t_j-\frac{[df/dt]_{t=t_j}}{[d^2f/dt^2]_{t=t_j}} \tag{4.6}$$

to the equation

$$\frac{d}{dt}f(t)=0, \tag{4.7}$$

to get the sequence $(t_1, t_2, \ldots)$ rapidly converging to t^*:

$$t_j \to t^* \qquad (j=1,2,\ldots). \tag{4.8}$$

5. Convergence. In the following we shall show that the sequence $(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots)$ converges to an optimum $\hat{\boldsymbol{x}}$ quadratically in the nondegenerate case at the final stages of iteration. To show this, we first note that the direction $\boldsymbol{\xi}^{(\nu)}$ determined by (4.1) is a "nice" direction. Let us denote $\boldsymbol{x}^{(\nu)} - \hat{\boldsymbol{x}}$ by $\boldsymbol{\varepsilon}^{(\nu)}$:

$$\varepsilon^{(\nu)\kappa} = x^{(\nu)\kappa} - \hat{x}^{\kappa}. \tag{5.1}$$

Before all, we see that

$$\begin{aligned} \sum_{\kappa=1}^{n} \tilde{c}_{\kappa}(\boldsymbol{x}^{(\nu)})\varepsilon^{(\nu)\kappa} &= \frac{c(\boldsymbol{x}^{(\nu)}) - c(\hat{\boldsymbol{x}})}{c(\boldsymbol{x}^{(\nu)})} = 1, \\ \sum_{\kappa=1}^{n} \tilde{a}^{i}_{\kappa}(\boldsymbol{x}^{(\nu)})\varepsilon^{(\nu)\kappa} &= \frac{a^i(\boldsymbol{x}^{(\nu)}) - a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})} = 1 - \frac{a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})} \,(\le 1). \end{aligned} \tag{5.2}$$

Then we have

$$\begin{aligned} \sum_{\kappa=1}^{n} \eta^{(\nu)}_{\kappa}\varepsilon^{(\nu)\kappa} &= (m+1) \sum_{\kappa=1}^{n} \tilde{c}_{\kappa}(\boldsymbol{x}^{(\nu)})\varepsilon^{(\nu)\kappa} - \sum_{i=1}^{m}\sum_{\kappa=1}^{n} \tilde{a}^{i}_{\kappa}(\boldsymbol{x}^{(\nu)})\varepsilon^{(\nu)\kappa} \\ &= (m+1) - m + \sum_{i=1}^{m} \frac{a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})} = 1 + \sum_{i=1}^{m} \frac{a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})} \,(\ge 1), \end{aligned} \tag{5.3}$$

$$\begin{aligned} \sum_{\kappa=1}^{n} H^{(\nu)}_{\lambda\kappa}\varepsilon^{(\nu)\kappa} &= -(m+1)\tilde{c}_{\lambda}(\boldsymbol{x}^{(\nu)}) \sum_{\kappa=1}^{n} \tilde{c}_{\kappa}(\boldsymbol{x}^{(\nu)})\varepsilon^{(\nu)\kappa} \\ &\quad + \sum_{i=1}^{m} \tilde{a}^{i}_{\lambda}(\boldsymbol{x}^{(\nu)}) \sum_{\kappa=1}^{n} \tilde{a}^{i}_{\kappa}(\boldsymbol{x}^{(\nu)})\varepsilon^{(\nu)\kappa} + \eta^{(\nu)}_{\lambda} \sum_{\kappa=1}^{n} \eta^{(\nu)}_{\kappa}\varepsilon^{(\nu)\kappa} \\ &= -(m+1)\tilde{c}_{\lambda}(\boldsymbol{x}^{(\nu)}) + \sum_{i=1}^{m} \left[\tilde{a}^{i}_{\lambda}(\boldsymbol{x}^{(\nu)}) \left(1 - \frac{a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})}\right)\right] \\ &\quad + \eta^{(\nu)}_{\lambda}\left(1 + \sum_{i=1}^{m} \frac{a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})}\right) \\ &= -\eta^{(\nu)}_{\lambda} - \sum_{i=1}^{m} \frac{a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})} \tilde{a}^{i}_{\lambda}(\boldsymbol{x}^{(\nu)}) + \eta^{(\nu)}_{\lambda} + \left(\sum_{i=1}^{m} \frac{a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})}\right)\eta^{(\nu)}_{\lambda} \\ &= \left(\sum_{i=1}^{m} \frac{a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})}\right)\eta^{(\nu)}_{\lambda} - \sum_{i=1}^{m} \frac{a^i(\hat{\boldsymbol{x}})}{a^i(\boldsymbol{x}^{(\nu)})} \tilde{a}^{i}_{\lambda}(\boldsymbol{x}^{(\nu)}). \end{aligned} \tag{5.4}$$

If we denote the set of active constraints at $\hat{\boldsymbol{x}}$ by

$$I = \{i \mid a^i(\hat{\boldsymbol{x}}) = 0\} \tag{5.5}$$

as in Section 3, we have from (5.4)

$$\sum_{\kappa=1}^{n} H_{\lambda\kappa}^{(\nu)} \varepsilon^{(\nu)\kappa} = w(x^{(\nu)})\left[\eta_\lambda^{(\nu)} - \sum_{i \notin I} \frac{w^i(x^{(\nu)})}{w(x^{(\nu)})} \tilde{a}_\lambda^i(x^{(\nu)})\right], \tag{5.6}$$

where

$$\begin{aligned} w^i(x) &= a^i(\hat{x})/a^i(x), \\ w(x) &= \sum_{i \notin I} w^i(x). \end{aligned} \tag{5.7}$$

(Note that $w^i(x) = 0$ for $i \in I$.) Comparing (4.1), (5.1), and (5.6), we might expect that, if $\boldsymbol{\varepsilon}^{(\nu)}$ is small enough,

$$y^{(\nu)} \equiv x^{(\nu)} + w(x^{(\nu)})\xi^{(\nu)} \tag{5.8}$$

would be a better approximation to $\hat{x}$. More rigorously, we can prove the quadratic convergence in the nondegenerate case.

Proposition 5.1. *Suppose that there is a unique optimum basis (i.e., neither primal nor dual degenerate) with the corresponding solution $\hat{x}$, to which $x^{(\nu)}$ converges. Then, if $\|x^{(\nu)} - \hat{x}\|$ is sufficiently small, $\|x^{(\nu+1)} - \hat{x}\| = O(\|x^{(\nu)} - \hat{x}\|^2)$, implying the quadratic convergence.*

Proof. By means of an affine transformation which moves $\hat{x}$ to $\mathbf{0}$ and makes the matrix a_κ^i into the basis form with respect to $\hat{x}$, the problem is expressed in the form

$$\min\left\{\sum_{\kappa=1}^{n} \bar{c}_\kappa x^\kappa \,\middle|\, \sum_{\kappa=1}^{n} \bar{a}_\kappa^i x^\kappa - \bar{a}_0^i \geq 0 (i = 1, \ldots, m)\right\}, \tag{5.9}$$

where $\bar{c}_\kappa > 0$ ($\kappa = 1, \ldots, n$) and $\bar{a}_\kappa^i = \delta_\kappa^i$, $\bar{a}_0^i = 0$ ($\kappa = 1, \ldots, n$; $i = 1, \ldots, n$; $\delta_\kappa^i = 1$ if $\kappa = i$ and $\delta_\kappa^i = 0$ otherwise), and $\bar{a}_0^i < 0$ ($i = n+1, \ldots, m$) and $\mathbf{0}$ is the unique optimum basic solution. We have only to prove that, in problem (5.9), for the νth approximate solution $x^{(\nu)}$, if $\|x^{(\nu)}\|$ is sufficiently small, $\|x^{(\nu+1)}\| = O(\|x^{(\nu)}\|^2)$.

Setting $\boldsymbol{\varepsilon} = x^{(\nu)}$, consider another problem scaled by $\boldsymbol{\varepsilon}$:

$$\min\left\{\sum_{\kappa=1}^{n} c_\kappa x^\kappa \,\middle|\, a^i(x) \equiv \sum_{\kappa=1}^{n} a_\kappa^i x^\kappa - a_0^i \geq 0 (i = 1, \ldots, m)\right\}, \tag{5.10}$$

where $c_\kappa = \bar{c}_\kappa \varepsilon^\kappa$, $a_\kappa^i = \bar{a}_\kappa^i \varepsilon^\kappa$, and $a_0^i = \bar{a}_0^i$. In order to prove the proposition, it suffices to prove that, in problem (5.10), for $x^{(\nu)} = e (e^\kappa = 1$ for $\kappa = 1, \ldots, n)$, we have $\|x^{(\nu+1)}\| = O(\|\boldsymbol{\varepsilon}\|)$.

Let us first note that $y \equiv y^{(\nu)}$ defined by (5.8) satisfies the equation

$$\sum_{\kappa=1}^{n} H_{\lambda\kappa}(e) y^\kappa = -\sum_{i=n+1}^{m} w^i(e) \frac{a_\lambda^i}{a^i(e)}.$$

The right-hand side of this equation is $O(\|\varepsilon\|)$. As we shall show in the following claim, the eigenvalues of $H_{\lambda\kappa}(e)$ are all of magnitude $\Theta(1)$, so that $\|y\| = O(\|\varepsilon\|)$.

Claim. When $\|\varepsilon\|$ is sufficiently small, all the eigenvalues $\sigma_1 \geq \cdots \geq \sigma_n > 0$ of $H_{\lambda\kappa}(e)$ is of order $\Theta(1)$.

Proof of Claim. From (3.9) we have

$$\begin{aligned}H_{\lambda\kappa}(e) &= \left(-m\frac{c_\lambda}{c(e)}+1\right)\left(-m\frac{c_\kappa}{c(e)}+1\right)+\sum_{i=1}^{n}\left(-\frac{c_\lambda}{c(e)}+\delta^i_\lambda\right)\left(-\frac{c_\kappa}{c(e)}+\delta^i_\kappa\right)\\ &\quad +(m-n)\frac{c_\lambda}{c(e)}\frac{c_\kappa}{c(e)}+O(\|\varepsilon\|)\\ &= \sum_{i=1}^{n+2} D^i_\lambda D^i_\kappa + O(\|\varepsilon\|),\end{aligned}$$

$$\begin{aligned}D^i_\kappa &= \delta^i_\kappa - \frac{c_\kappa}{c(e)} && (i=1,\ldots,n)\\ &= 1-m\frac{c_\kappa}{c(e)} && (i=n+1)\\ &= \sqrt{m-n}\,\frac{c_\kappa}{c(e)} && (i=n+2).\end{aligned}$$

When $\|\varepsilon\|$ is sufficiently small, we may apply the Binet-Cauchy formula to the principal term of the above expression for $H_{\lambda\kappa}(e)$ to get

$$\begin{aligned}\prod_{\kappa=1}^{n}\sigma_\kappa &= |\det H_{\lambda\kappa}(e)|\\ &= \left[(m-n)(m-n+1)\sum_{\kappa=1}^{n}\left(\frac{c_\kappa}{c(e)}\right)^2+\sum_{\lambda<\kappa}(m-n)\left(\frac{c_\lambda-c_\kappa}{c(e)}\right)^2\right][1+O(\|\varepsilon\|)]\\ &\geq \frac{(m-n)(m-n+1)}{n}[1+O(\|\varepsilon\|)],\end{aligned}$$

where we made use of the fact that $\sum_{\kappa=1}^{n} c_\kappa/c(e) = 1$.

Furthermore, we have

$$\begin{aligned}\sum_{\kappa=1}^{n}\sigma_\kappa &= \|D^i_\kappa\|^2_E[1+O(\|\varepsilon\|)]\\ &= \sum_{i=1}^{n+2}\sum_{\kappa=1}^{n}|D^i_\kappa|^2[1+O(\|\varepsilon\|)]\\ &= \left\{m(m+1)\sum_{\kappa=1}^{n}\left(\frac{c_\kappa}{c(e)}\right)^2-2(m-n+1)\right\}[1+O(\|\varepsilon\|)]\\ &\leq [m(m+1)-2(m-n+1)][1+O(\|\varepsilon\|)]\\ &= [m(m-1)+2(n-1)][1+O(\|\varepsilon\|)].\end{aligned}$$

Since

$$\sum_{\kappa=1}^{n} \sigma_\kappa \geq \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n = \prod_{\kappa=1}^{n} \sigma_\kappa \Big/ \prod_{\kappa=1}^{n-1} \sigma_\kappa \geq (n-1)^{n-1} \prod_{\kappa=1}^{n} \sigma_\kappa \Big/ \Big(\sum_{\kappa=1}^{n} \sigma_\kappa \Big)^{n-1}$$

and

$$(n-1)^{n-1} \prod_{\kappa=1}^{n} \sigma_\kappa \Big/ \Big(\sum_{\kappa=1}^{n} \sigma_\kappa \Big)^{n-1} \geq \frac{(n-1)^{n-1}(m-n)(m-n+1)}{n[m(m-1)+2(n-1)]^{n-1}} [1 + O(\|\boldsymbol{\varepsilon}\|)],$$

the claim follows. (End of Proof of the Claim.)

Let z be the first point on a constraint hyperplane $a^i(x) = 0$ $(i = 1, \ldots, n)$ that is met by a ray emanating from $\boldsymbol{e}$ toward $\boldsymbol{y}$ in the direction of $\boldsymbol{\xi}^{(\nu)}$. Since $\|\boldsymbol{y}\| = O(\|\boldsymbol{\varepsilon}\|)$, we have $\|z\| = O(\|\boldsymbol{\varepsilon}\|)$. The line search over the line segment connecting $\boldsymbol{e}$ and z is to find the minimum point τ^* of

$$f(\tau) = c(x)^{m+1} \Big/ \prod_{i=1}^{m} a^i(x),$$

where $x = \tau e + (1-\tau) z$ $(0 \leq \tau \leq 1)$. Setting $\gamma = \sum_{\kappa=1}^{n} c_\kappa (e^\kappa - z^\kappa)$ and $\alpha^i = \sum_{\kappa=1}^{n} a^i_\kappa (e^\kappa - z^\kappa)$, we have

$$\frac{1}{f(\tau)} \frac{df}{d\tau} = \frac{(m+1)\gamma}{c(z) + \gamma\tau} - \sum_{i=1}^{m} \frac{\alpha^i}{a^i(z) + \alpha^i \tau}$$
$$\geq \frac{(m+1)\gamma}{c(z) + \gamma\tau} - \frac{m}{\tau}.$$

We note that $(m+1)\gamma/[c(z) + \gamma\tau] - m/\tau = 0$ when $\tau = \tau' \equiv mc(z)/\gamma$. Thus, if $\|\varepsilon\|$ is sufficiently small, we have $\tau' = O(\|\varepsilon\|)$ and $(df/d\tau)|_{\tau=\tau'} \geq 0$. From the convexity of $f(\tau)$, $\tau^* \leq \tau'$. Thus, we have $\tau^* = O(\|\varepsilon\|)$, and hence $\|x^{(\nu+1)}\| = O(\|\boldsymbol{\varepsilon}\|)$ for $x^{(\nu+1)} = \tau^* e + (1-\tau^*) z$. □

In the degenerate case $\hat{x}$ is optimum but may not be basic. In that case $H^{(\nu)}_{\lambda\kappa}$ multiplied by $c(x^{(\nu)})^2$ may have small eigenvalues. However, the corresponding eigenvectors are seen to be nearly parallel to $\hat{X}$, so that, as far as the component orthogonal to $\hat{X}$ are concerned, we can obtain a conclusion similar to the above.

It is also seen from (5.7) and (5.8) that, if $x^{(\nu)}$ is close enough to $\hat{x}$, the t^* determined by the line search in the algorithm is expected to be nearly equal to

$$w(x^{(\nu)}) = \sum_{i \notin I} w^i(x^{(\nu)}) = \sum_{i \notin I} \frac{a^i(\hat{x})}{a^i(x^{(\nu)})} \approx m - |I| \equiv \hat{t}. \tag{5.11}$$

(See (5.5) for the definition of I. Note that $|I| = n$ in the nondegenerate case.)

When $x^{(\nu)}$ is not sufficiently close to $\hat{x}$, we cannot expect a superlinear convergence, but, at best, only a linear one. The following is not a rigorous proof of the linear convergence, but suggests a theoretical explanation for the empirical behavior of the algorithm, i.e., the linear convergence, observed in the computational experiments, and gives an intuitive sketch on which a more formal proof may be constructed.

PROPOSITION 5.2. *Suppose that, in an arbitrary direction $\boldsymbol{\xi}$ starting from $\boldsymbol{x}^{(\nu)}$, the value of the function $F(\boldsymbol{x}^{(\nu)}+t\boldsymbol{\xi})$ can be sufficiently well approximated as*

$$F(\boldsymbol{x}^{(\nu)}+t\boldsymbol{\xi}) \approx F(\boldsymbol{x}^{(\nu)})+t\sum_{\kappa=1}^{n}\xi^{\kappa}\frac{\partial}{\partial x^{\kappa}}F(\boldsymbol{x}^{(\nu)})+\frac{t^2}{2}\sum_{\kappa=1}^{n}\sum_{\lambda=1}^{n}\xi^{\kappa}\xi^{\lambda}\frac{\partial^2}{\partial x^{\lambda}\partial x^{\kappa}}F(\boldsymbol{x}^{(\nu)})$$

for t satisfying the inequality

$$\frac{t^2}{2}\sum_{\kappa=1}^{n}\sum_{\lambda=1}^{n}H_{\lambda\kappa}^{(\nu)}\xi^{\kappa}\xi^{\lambda}\le K^2 \tag{5.12}$$

with an appropriate value of K.[6] *Then,*

$$\frac{F(\boldsymbol{x}^{(\nu+1)})}{F(\boldsymbol{x}^{(\nu)})}<1-K(1-K).$$

PROOF. Under the assumption, we have

$$\frac{f(t)}{f(0)}=\frac{F(\boldsymbol{x}^{(\nu)}+t\boldsymbol{\xi})}{F(\boldsymbol{x}^{(\nu)})}\approx 1+t\sum_{\kappa=1}^{n}\eta_{\kappa}^{(\nu)}\xi^{\kappa}+\frac{t^2}{2}\sum_{\kappa=1}^{n}\sum_{\lambda=1}^{n}H_{\lambda\kappa}^{(\nu)}\xi^{\kappa}\xi^{\lambda}$$

as long as t satisfies (5.12). This means that $f(t)/f(0)$ can be approximated by the linear function in t

$$1+t\sum_{\kappa=1}^{n}\eta_{\kappa}^{(\nu)}\xi^{\kappa} \tag{5.13}$$

up to an error less than K^2. We consider the problem of minimizing (5.13) under condition (5.12), for which the solution is obvious:

$$t=\sqrt{2K^2\Big/\sum_{\kappa=1}^{n}\sum_{\lambda=1}^{n}H_{\lambda\kappa}^{(\nu)}\xi^{\kappa}\xi^{\lambda}} \tag{5.14}$$

if $\sum_{\kappa=1}^{n}\eta_{\kappa}^{(\nu)}\xi^{\kappa}<0$. The minimum value of (5.13) depends on $\boldsymbol{\xi}$, and, as is readily seen, the minimum of that minimum value with $\boldsymbol{\xi}$ varied is attained if $\boldsymbol{\xi}$ is chosen equal to $\boldsymbol{\xi}^{(\nu)}$ determined by (4.1).

By determining $\boldsymbol{\xi}^{(\nu)}$ and t in that manner, we shall get

$$\begin{aligned}\hat{r}=\frac{f(t)}{f(0)}&\approx 1+t\sum_{\kappa=1}^{n}\eta_{\kappa}^{(\nu)}\xi^{(\nu)\kappa}+K^2\\&=1-K\sqrt{2\sum_{\kappa=1}^{n}\sum_{\lambda=1}^{n}H_{\lambda\kappa}^{(\nu)}\xi^{(\nu)\kappa}\xi^{(\nu)\lambda}}+K^2.\end{aligned} \tag{5.15}$$

[6] This assumption may seem too strict in the general situation, and is, indeed, sometimes hard to hold valid in all directions. However, as is seen from the proof, it suffices for the assumption to hold only for the directions $\boldsymbol{\xi}^{(\nu)}$ and $\hat{\boldsymbol{x}}-\boldsymbol{x}^{(\nu)}$, which, in experiments, is observed almost always to be the case.

If we chose $\boldsymbol{\xi}=\hat{\boldsymbol{x}}-\boldsymbol{x}^{(\nu)}$ $(=-\boldsymbol{\varepsilon}^{(\nu)})$ and determined t by (5.14) accordingly, we should have

$$\sum_{\kappa=1}^{n} \eta_{\kappa}^{(\nu)}\xi^{\kappa}=-1-w(\boldsymbol{x}^{(\nu)}),$$

$$\begin{aligned}\sum_{\kappa=1}^{n}\sum_{\lambda=1}^{n} H_{\lambda\kappa}^{(\nu)}\xi^{\kappa}\xi^{\lambda} &= w(\boldsymbol{x}^{(\nu)})\left[1+w(\boldsymbol{x}^{(\nu)})+\sum_{i\notin I}\frac{w^{i}(\boldsymbol{x}^{(\nu)})}{w(\boldsymbol{x}^{(\nu)})}(-1+w^{i}(\boldsymbol{x}^{(\nu)}))\right]\\ &= w(\boldsymbol{x}^{(\nu)})+w(\boldsymbol{x}^{(\nu)})^{2}-w(\boldsymbol{x}^{(\nu)})+\sum_{i\notin I} w^{i}(\boldsymbol{x}^{(\nu)})^{2}\\ &= w(\boldsymbol{x}^{(\nu)})^{2}+\sum_{i\notin I} w^{i}(\boldsymbol{x}^{(\nu)})^{2}\end{aligned}$$

(cf. (5.2), (5.3), (5.6), and (5.7)), or

$$(5.16)\qquad \begin{aligned} r=\frac{f(t)}{f(0)} &\approx 1-t(1+w(\boldsymbol{x}^{(\nu)}))+K^{2}\\ &=1-\frac{\sqrt{2}K(1+w(\boldsymbol{x}^{(\nu)}))}{\sqrt{w(\boldsymbol{x}^{(\nu)})^{2}+\sum_{i\notin I} w^{i}(\boldsymbol{x}^{(\nu)})^{2}}}+K^{2}.\end{aligned}$$

Since $w^{i}(\boldsymbol{x}^{(\nu)})>0$ and $w(\boldsymbol{x}^{(\nu)})=\sum_{i\notin I} w^{i}(\boldsymbol{x}^{(\nu)})$, we have

$$\sum_{i\notin I} w^{i}(\boldsymbol{x}^{(\nu)})^{2}\leq w(\boldsymbol{x}^{(\nu)})^{2},$$

so that we have

$$(5.17)\qquad \sqrt{w(\boldsymbol{x}^{(\nu)})^{2}+\sum_{i\notin I} w_{i}(\boldsymbol{x}^{(\nu)})^{2}}\leq\sqrt{2}w(\boldsymbol{x}^{(\nu)}),$$

or

$$(5.18)\qquad r\leq 1-\frac{K(1+w(\boldsymbol{x}^{(\nu)}))}{w(\boldsymbol{x}^{(\nu)})}+K^{2}=1-K-\frac{K}{w(\boldsymbol{x}^{(\nu)})}+K^{2}.$$

Since the $\hat{r}$ in (5.15) is by definition not greater than the r in (5.18), we finally have

$$(5.19)\qquad \hat{r}<1-K(1-K).\qquad\square$$

Thus, if the assumption of Proposition 5.2 is satisfied, at each iteration step, we have the value of the function $F(\boldsymbol{x}^{(\nu)})$ reduced at least by a constant factor $1-K(1-K)$ approximately:

$$(5.20)\qquad \frac{F(\boldsymbol{x}^{(\nu+1)})}{F(\boldsymbol{x}^{(\nu)})}<1-K(1-K),$$

the factor being equal, for example, to $\frac{3}{4}$, $\frac{13}{16}$, and $\frac{57}{64}$ with $K=\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$, respectively.

6. Some Remarks

6.1. Practical Efficiency of the Algorithm. The proposed algorithm seems to require a small number of iterations in practice. However, it obviously has the disadvantage that, at each iteration step, we have to solve a large, rather dense system of linear equations (4.1). In order for this algorithm to become "practical" in the true sense of the word, that disadvantage should be overcome. The prospect is not very gloomy but rather bright, because the gradient η_κ as well as the Hessian $H_{\lambda\kappa}$ has a seemingly very nice structure (see (3.4), (3.9), etc.). In fact, Professor Kunio Tanabe of the Institute of Statistical Mathematics suggested the possibility of applying a conjugate-gradient-type algorithm for the solution of the system of linear equations. Evidently there will also be many other devices to make this stage of computation more and more effective, e.g., using an approximate Hessian instead of the complete one, devising an effective way of updating the inverse (or, more practically, the LU- or the QR-decomposition) of the Hessian, etc., by making use of the special structures of the gradient and the Hessian.[7]

6.2. A Linear Subiteration Scheme. It may be possible to make use of some factorization (e.g., LU-decomposition) of a Hessian $H^{(\nu)}_{\lambda\kappa}$ more than once to improve an approximate solution $x^{(\nu)}$. In fact, from (5.6), we might expect the $x^{(\nu+1,1)}$ determined by the line search along $\xi^{(\nu,1)}$:

$$x^{(\nu+1,1)}=x^{(\nu)}+t\xi^{(\nu,1)},$$

where $\xi^{(\nu,1)}$ is determined by

$$(6.1)\qquad \sum_{\kappa=1}^{n} H^{(\nu)}_{\lambda\kappa}\xi^{(\nu,1)\kappa}=-\left(\sum_{i=1}^{m}\frac{a^i(x^{(\nu+1)})}{a^i(x^{(\nu)})}\right)\eta_\lambda(x^{(\nu)})+\sum_{i=1}^{m}\frac{a^i(x^{(\nu+1)})}{a^i(x^{(\nu)})}\,\tilde{a}^i_\lambda(x^{(\nu)}),$$

would be a better approximation than $x^{(\nu)}$. We may repeat this process by regarding $x^{(\nu+1,1)}$ as $x^{(\nu+1)}$. Computational experiments have shown that it is sometimes, though not very often, the case. Since, once we have the LU-decomposition of $H^{(\nu)}_{\lambda\kappa}$, iteration (6.1) is not very costly, we may iterate (6.1) as long as the value of $F(x^{(\nu,l)})$ is being improved substantially.

[7] A referee has kindly informed the authors that the following recent report contains such devices: P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright, "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method", Technical Report SOL 85-11, Department of Operations Research, Stanford University, 1985.

6.3. An Extension. Dr. Kazuo Murota of the University of Tsukuba reported, in a personal communication to the authors, his observation that the function $F(x)$ defined in (3.1) remains convex if we take any "convex" function for $c(x)$ and any "concave" functions for the $a^i(x)$'s. In fact, we then have the same expression (3.6) for $\eta_\kappa(x)=[1/F(x)](\partial F(x)/\partial x^\kappa)$ if we replace the definitions of $\tilde{c}_\kappa(x)$ and the $\tilde{a}^i_\kappa(x)$'s by

$$\tag{6.2} \begin{aligned} \tilde{c}_\kappa(x) &= \frac{1}{c(x)}\frac{\partial}{\partial x^\kappa}c(x), \\ \tilde{a}^i_\kappa(x) &= \frac{1}{a^i(x)}\frac{\partial}{\partial x^\kappa}a^i(x), \\ \bar{a}_\kappa(x) &= \frac{1}{m}\sum_{i=1}^m \tilde{a}^i_\kappa(x). \end{aligned}$$

$B_{\lambda\kappa}=\partial^2 \log F(x)/\partial x^\lambda \partial x^\kappa$ is expressed as

$$\tag{6.3} \begin{aligned} B_{\lambda\kappa}(x) &= \left\{\frac{m+1}{c(x)}\frac{\partial^2}{\partial x^\lambda \partial x^\kappa}c(x) - \sum_{i=1}^m \frac{1}{a^i(x)}\frac{\partial^2}{\partial x^\lambda \partial x^\kappa}a^i(x)\right\} \\ &\quad + \left\{-(m+1)\tilde{c}_\lambda(x)\tilde{c}_\kappa(x) + \sum_{i=1}^m \tilde{a}^i_\lambda(x)\tilde{a}^i_\kappa(x)\right\}, \end{aligned}$$

and (3.8) remains unchanged. Since $c(x)$ is convex and the $a^i(x)$'s are concave, the matrix in the first pair of braces on the right-hand side of (6.3) is nonnegative definite. Furthermore, as is easily seen, the manipulation of completing the squares for the sum of the second pair of braces on the right-hand side of (6.3) and $\eta_\lambda(x)\eta_\kappa(x)$ can be carried out in entirely the same manner as in (3.9). Therefore $H_{\lambda\kappa}(x)$ is still positive definite in this generalized case.

It is not quite obvious whether this observation of Murota's is of some direct significance for a nonlinear programming problem or not, but, in the context of this paper, it shows us the interesting fact that $F(x)=c(x)^{m'}/\prod_{i=1}^m a^i(x)$ with any m' greater than or equal to $m+1$ is always convex in the case of linear programming. To prove this fact, we may simply replace $c(x)$ by $c'(x)=c(x)^{m'/(m+1)}$, the latter being convex if $c(x)$ is linear.

6.4. Observations on Invariance. It is one of the principal characteristics and advantages of our algorithm that, except for the arbitrariness of the choice of an initial feasible point, everything is invariant under (i) the group of affine transformations of $\mathbf{R}^n$ and (ii) the group of rescalings of the functions $a^i(x)$ and $c(x)$ (where "rescaling" means multiplication of each $a^i(x)$ (or $c(x)$) by an arbitrary constant). This makes a clear contrast with Karmarkar's algorithm which requires the introduction of an artificial constraint $\sum x^\kappa = M$ which is not affine-invariant. Of course, Karmarkar's algorithm enjoys another type of invariance, i.e., the invariance under the projective transformations in another space, as is well known [4].

Part II. Preliminary Experiments

7. Design of Computational Experiments. We show the results of preliminary computational experiments of the algorithm proposed in Part I. The algorithm was coded into a FORTRAN program for the purpose of investigating the behavior of the algorithm only; therefore the advanced techniques discussed in Section 6 were not made use of. All the experiments were executed with double-precision (14 hexadecimal digits) arithmetic on the HITAC M-280H of the Computer Centre of the University of Tokyo.

It is now widely known that the timing data are considerably dependent on the implementation technique as well as on the particular architecture of the machine. However, the results we show in the following, i.e., how fast the function value $F(x^{(\nu)})$ decreases as the iteration proceeds, how the number of necessary iterations depends on the problem size, etc., will be almost independent of the implementation and of the machine arthitecture. Here, it is important to note that for an algorithm which produces a "linearly convergent" sequence (as Khachian's and also Karmarkar's[8]), the number of necessary iterations is highly dependent on the required accuracy, whereas for an algorithm producing a "superlinearly convergent" sequence (as ours), it is insensitive to the required accuracy.

The computational experiments were performed first on small-size sample problems and then on three classes of small- and medium-size structured problems with parameters. Most of the small-size problems were of two variables ($n=2$) (Examples ⟨C1⟩-⟨C6⟩), for which the feasible regions as well as the histories of iterative processes of the algorithm are here illustrated together with the contour lines of the barrier functions $F(\boldsymbol{x})$. A "diet problem" (Example ⟨D⟩) was also taken as a small-size sample problem. The three classes of the structured problems are the Klee-Minty problems [7] (Example ⟨KM⟩), the random problems (Example ⟨R⟩), and the assignment problems (Example ⟨A⟩). The sizes of these problems can be varied by adjusting the parameters, which enables us to know what effect the size of the problem has on the performance of the algorithm. We also show by computational experiments how effectively the linear subiteration scheme given in Section 6.2 works.

In order to make the value of the objective function for the optimum solution equal to zero and to produce an initial interior feasible solution we adopted the following primal-dual pair approach. Consider the linear programming problem:

$$\begin{aligned} &\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}, \\ &A\boldsymbol{x} \geq \boldsymbol{b}, \qquad (7.1) \\ &\boldsymbol{x} \geq 0. \end{aligned}$$

[8] In many computational experiments, it has been observed that Karmarkar's algorithm also produces a linearly convergent sequence. It seems that no theoretical proof has been done for the superlinear convergence of Karmarkar's algorithm.

From the duality theorem, this problem is equivalent to finding a feasible solution of the following linear inequalities:

$$\begin{aligned} Ax &\geq b, \\ -A^{\mathrm{T}}y &\geq -c, \\ -c^{\mathrm{T}}x + b^{\mathrm{T}}y &\geq 0, \\ x, y &\geq 0. \end{aligned} \tag{7.2}$$

Rewrite (7.2) by $\tilde{A}\tilde{x} \geq \tilde{a}_0$ and $\tilde{x} \geq 0$ with $\tilde{x} = (x^{\mathrm{T}}, y^{\mathrm{T}})^{\mathrm{T}}$. In order to find a feasible solution of (7.2), let us consider the following linear programming problem in the canonical form (2.1) and (2.2) of Part I:

$$\begin{aligned} &\min \lambda, \\ &\tilde{A}\tilde{x} + \tilde{a}\lambda - \tilde{a}_0 \geq 0, \\ &\tilde{x}, \lambda \geq 0, \end{aligned} \tag{7.3}$$

where $\tilde{a}$ is defined as follows: let $e = (1, \ldots, 1)^{\mathrm{T}}$ and $a' = \tilde{A}e - \tilde{a}_0$; then set $\tilde{a}_i = 1 - a_i'$ if $a_i' \leq 0$ and $\tilde{a}_i = 0$ if $a_i' > 0$. Then, $\tilde{x}^{(0)} = e$, $\lambda^{(0)} = 1$ is an initial interior feasible solution for (7.3), and (7.1) has the optimum solution iff the value of the objective function for the optimum solution of (7.3) is zero; furthermore, the optimum solution of (7.3) with the value of the objective function equal to zero affords the optimal pair of primal and dual solutions for (7.1).

Throughout the following computational experiments, we did not adopt any theoretical stopping criteria for the iteration, but we continued the iteration until the finite-precision computation would proceed no further. As for the validity of the assumptions in Section 2, we were "optimistic," i.e., without checking their validity for each problem we entered into the computation to see what would happen.

8. Small-Size Problems.

Examples ⟨C1⟩-⟨C6⟩: We first consider six problems with two variables $x = (x, y)$ $(n = 2)$ for which the values of objective functions for the optimum solutions are *a priori* known to be zero. The computational experiments were executed with three starting points chosen rather arbitrarily for each problem. The problems are as follows, where $c = c(x)$ and $a^i = a^i(x)$:

⟨C1⟩: $c = x + y$; $a^1 = x$; $a^2 = y$; $a^3 = 2 - 2x - y$; $a^4 = 3 + 2x - 4y$.

⟨C2⟩: $c = x + y$; $a^1 = x$; $a^2 = y$; $a^3 = 2 - 2x - y$; $a^4 = 3 + 2x - 4y$; $a^5 = x + 2y$.

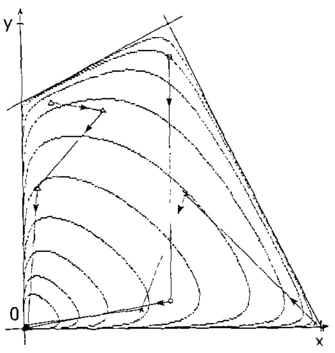
⟨C3⟩: $c = 3 + 2x - 4y$; $a^1 = x$; $a^2 = y$; $a^3 = 2 - 2x - y$; $a^4 = 3 + 2x - 4y$.

⟨C4⟩: $c = 1 - x$; $a^1 = x$; $a^2 = y$; $a^3 = 1 - x - y$.

⟨C5⟩: $c = x - y/4 + \frac{1}{12}$; $a^1 = x$; $a^2 = y$; $a^3 = x + y - \frac{1}{3}$; $a^4 = 2x - y + \frac{1}{3}$; $a^5 = -3x + y + 2$; $a^6 = -x/4 - y + 1$.

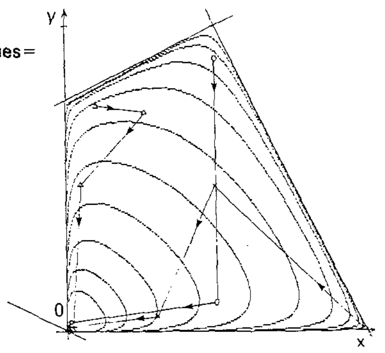
⟨C6⟩: $c = -x + y + \frac{2}{3}$; $a^1 = x$; $a^2 = y$; $a^3 = x + y - \frac{1}{3}$; $a^4 = 2x - y + \frac{1}{3}$; $a^5 = -3x + y + 2$; $a^6 = -x/4 - y + 1$.

Function values=
1024.
256.
64.
16.
4.
1.
.25
.0625
.015625
.00390625



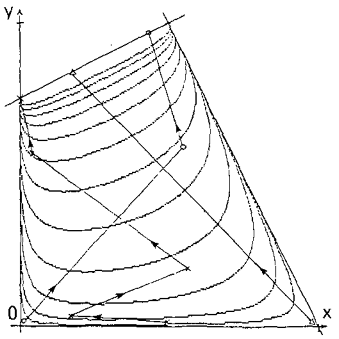
(CONTOUR1)
c=x+y; a(1)=x; a(2)=y; a(3)=2.0−2.0*x−y;
a(4)=3.0+2.0*x−4.0*y
(a)

Function values=
1024.
256.
64.
16.
4.
1.
.25
.0625
.015625
.00390625



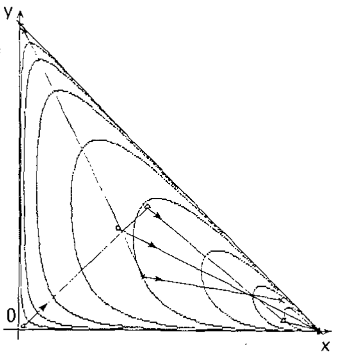
(CONTOUR2)
c=x+y; a(1)=x; a(2)=y; a(3)=2.0−2.0*x−y;
a(4)=3.0+2.0*x−4.0y; a(5)=x+2.0*y
(b)

Function values=
65536.
16384.
4096.
1024.
256.
64.
16.
4.
1.
.25
.0625
.015625



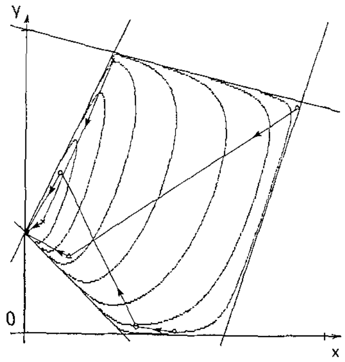
(CONTOUR3)
c=3.0+2.0*x−4.0y; a(1)=x; a(2)=y; a(3)=2.0−2.0*x−y;
a(4)=3.0+2.0*x−4.0*y
(c)

Function values=
1024.
256.
64.
16.
4.
1.
.25
.0625



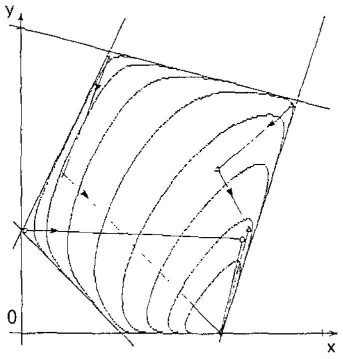
(CONTOUR4)
c=1.0−x; a(1)=x; a(2)=y; a(3)=1.0−x−y
(d)

Function values=
50.
5.
.5
.05
.005
.0005
.00005



(CONTOUR5)
c=x−y/4+1/12; a(1)=x; a(2)=y; a(3)=x+y−1/3;
a(4)=2*x−y+1/3; a(5)=−3*x+y+2; a(6)=−x/4−y+1
(e)

Function values=
10000.
1000.
100.
10.
1.
.1
.01
.001



(CONTOUR6)
c=−x+y+2/3; a(1)=x; a(2)=y; a(3)=x+y−1/3;
a(4)=2*x−y+1/3; a(5)=−3*x+y+2; a(6)=−x/4−y+1
(f)

Fig. 1. Computational results for the small sample problems with two variables. (a) Example ⟨C1⟩. (b) Example ⟨C2⟩. (c) Example ⟨C3⟩. (d) Example ⟨C4⟩. (f) Example ⟨C5⟩. (e) Example ⟨C6⟩.

We illustrate the process of iteration together with the contour lines of the barrier functions $F(x)$ in Figure 1 (from the contour lines, the good properties of the barrier functions would be observed visually).

Example ⟨D⟩: Next, we consider the following problem, which is a so-called diet problem, taken from the HITAC MPSII manual:

$$\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}, \qquad \begin{cases} A\boldsymbol{x} \ge \boldsymbol{b}, \\ \boldsymbol{x} \ge 0, \end{cases}$$

where

$$A = \begin{pmatrix} 351 & 270 & 260 & 451 & 156 & 59 & 721 & 58 & 130 & 118 & 77 & 51 & 40 & 24 & 28 & 311 & 40 \\ 6.2 & 8 & 17.5 & 12 & 12.7 & 2.9 & 0.6 & 6 & 17.5 & 20 & 1.9 & 1.3 & 1.2 & 1.6 & 3 & 34.2 & 0.8 \\ 0.8 & 1.5 & 20.5 & 44.3 & 11.2 & 3.3 & 81.6 & 3.5 & 6 & 3.5 & 0.1 & 0.2 & 0.2 & 0.2 & 0.4 & 0.7 & 0.3 \\ 6 & 11 & 6 & 9 & 65 & 100 & 10 & 120 & 80 & 12 & 5 & 35 & 40 & 45 & 98 & 470 & 14 \\ 2 & 480 & 90 & 90 & 90 & 36 & 780 & 5 & 100 & 90 & 12 & 57 & 10 & 15 & 25 & 600 & 4 \\ 0.4 & 1 & 1.3 & 1.2 & 2.6 & 0.1 & 0.1 & 1.4 & 3 & 0.7 & 0.5 & 0.5 & 0.5 & 0.4 & 3.3 & 23 & 0.2 \\ 0 & 0 & 25 & 0 & 800 & 120 & 2400 & 0 & 60 & 40 & 0 & 1300 & 6 & 33 & 2600 & 10000 & 40 \\ 0.09 & 0.1 & 0.04 & 0.4 & 0.1 & 0.04 & 0.01 & 0.02 & 0.02 & 0.15 & 0.1 & 0.06 & 0.03 & 0.08 & 0.12 & 0.21 & 0.09 \\ 0.03 & 0.03 & 0.11 & 0.1 & 0.4 & 0.15 & 0.03 & 0.02 & 0.15 & 0.2 & 0.03 & 0.04 & 0.02 & 0.05 & 0.3 & 1 & 0.02 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 1 & 15 & 7 & 10 & 50 & 100 & 20 & 50 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 530 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$\boldsymbol{b} = (2300, 75, 38, 660, 1300, 10, 1900, 1.2, 1.2, 63, 400)^{\mathrm{T}}$,

$\boldsymbol{c} = (25, 30, 350, 150, 40, 20, 100, 40, 60, 100, 17, 20, 20, 12, 75, 900, 20)^{\mathrm{T}}$.

The optimum value of this problem is 354.03042. We solved this problem in two ways: first, solving it directly with $c_0 = 354.03042$ and an initial interior feasible solution $(100, 100, \ldots, 100)^{\mathrm{T}}$, and second, solving the extended problem as described in (7.2) and (7.3). The results are shown in Figure 2.

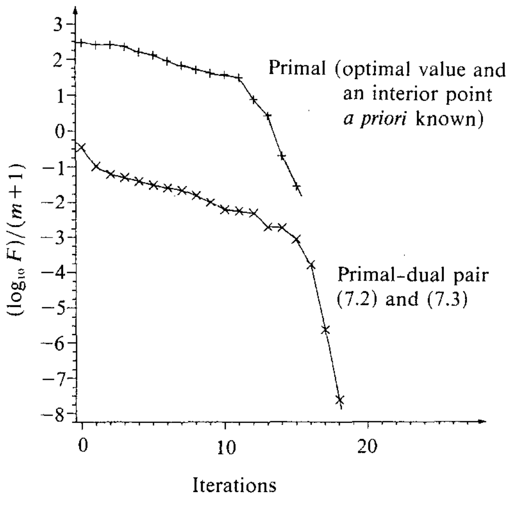


Fig. 2. Computational results for Example ⟨D⟩ (diet problem).

9. Klee–Minty Problems. Example ⟨KM⟩: The class of problems initially proposed by Klee and Minty [7] (the present form (9.1) is due to Avis and Chvátal [1]) is well known as linear programming problems with N variables for which the simplex method with various pivot rules (e.g., Bland's rule) requires an exponential, in N, number of pivot steps to reach the optimum:

$$\max \sum_{j=1}^{N} \varepsilon^{N-j} x_j,$$

$$2\sum_{j=1}^{i-1} \varepsilon^{i-j} x_j + x_i \leq 1 \qquad (i = 1, 2, \ldots, N), \tag{9.1}$$

$$x_j \geq 0 \qquad (j = 1, 2, \ldots, N),$$

where $0 < \varepsilon < \frac{1}{2}$. The optimum solution of this problem is $x_j = 0$ $(j = 1, \ldots, N-1)$ and $x_N = 1$. Concerning the size m, n of the problem obtained by combining this problem with its dual, $m = 2(2N+1)$ and $n = 2N+1$. Computational experiments were performed for the cases with $\varepsilon = 0.4$ and $N = 2$, 4, 8, 16. The results are shown in Figure 3.

Among the problems tested, the growth of the number of iterations required by the Klee–Minty problems as the number of variables increased was the most remarkable, but it seems still $O(N)$.

10. Random Problems. Example ⟨R⟩: Random linear programming problems were used by several computational experiments for the simplex algorithm (e.g.,

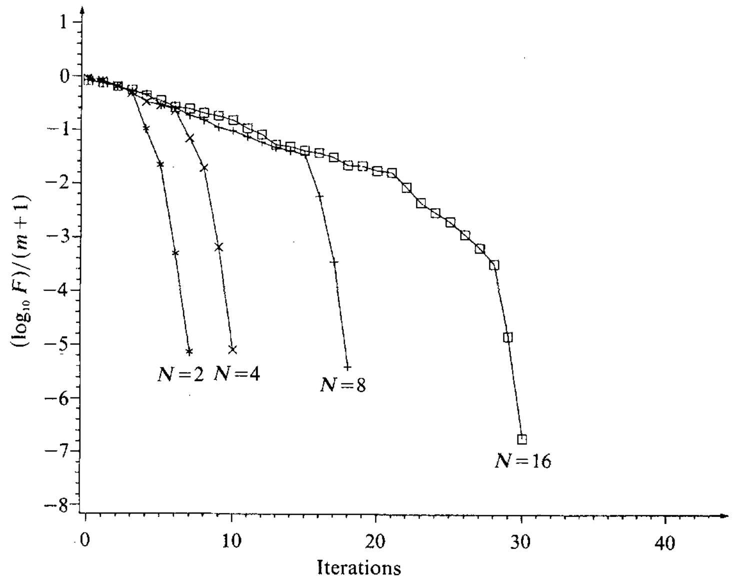


Fig. 3. Computational results for Example ⟨KM⟩ (the Klee–Minty problem).

see Avis and Chvátal [1]). The problem is described as follows:

$$\text{(10.1)} \qquad \max e^{\mathrm{T}} x, \quad \begin{cases} Ax \leq 10^4, \\ x \geq 0, \end{cases}$$

where $e = (1, \ldots, 1)^{\mathrm{T}}$ and A is an $M \times N$ matrix with $M \leq N$, the elements of which are random integers from 1 to 1000. The problems have many redundant constraints. Furthermore, it seems that problems of this class are not as practical since their A's are dense while the A's in the ordinary linear programming problems being solved in practice are sparse. However, the random problems give us some insight into the performance of the algorithm for dense problems, especially into the effect of problem size on the number of iterations required by the algorithm for structured dense problems.

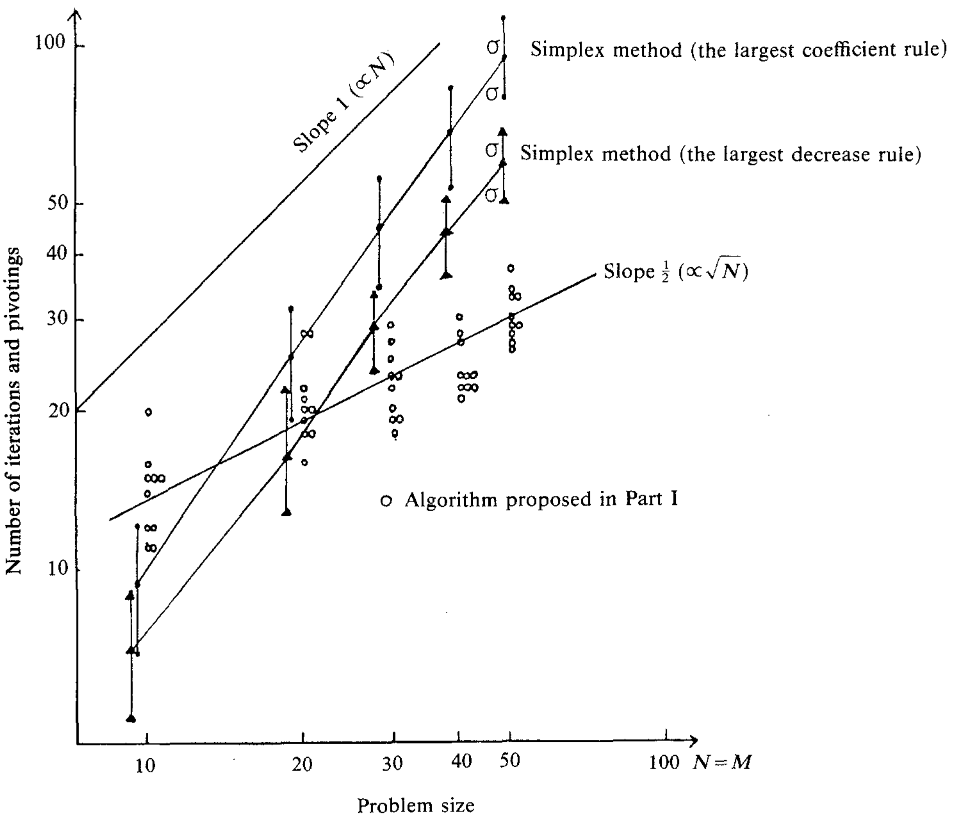


Fig. 4. Computational results for Example ⟨R⟩ (random problems): the number of iterations (○) and pivotings (▲, ●) required [1].

Computational experiments were executed for the cases with $M=N$ and $N=10, 20, 30, 40, 50$, where the size parameters m, n of the problem constructed as in (7.2) and (7.3) are $m=2(2N+1)$ and $n=2N+1$, where, in these computational experiments, each row of the original problem (10.1) is scaled, i.e., multiplied by a factor 10^{-4}. The results are shown in Figures 4 and 5.

In Figure 4 we also depict the results, taken from Avis and Chvátal [1], on the number of pivotings of the simplex methods for the original problems (not the primal-dual problems). It is seen that the simplex method requires about $O(N^{1.5})$ pivotings, while the algorithm proposed in Part I requires about $O(n^{0.5})$ iterations. (Note that one pivoting in the simplex method is cheaper than one iteration of the proposed algorithm with a naive implementation, and so it is not fair to compare the two algorithms based on these results alone.) If one iteration of our algorithm is to be compared to M pivotings of the simplex algorithm, then the two algorithms may have the same performance, at least in the order of magnitude.

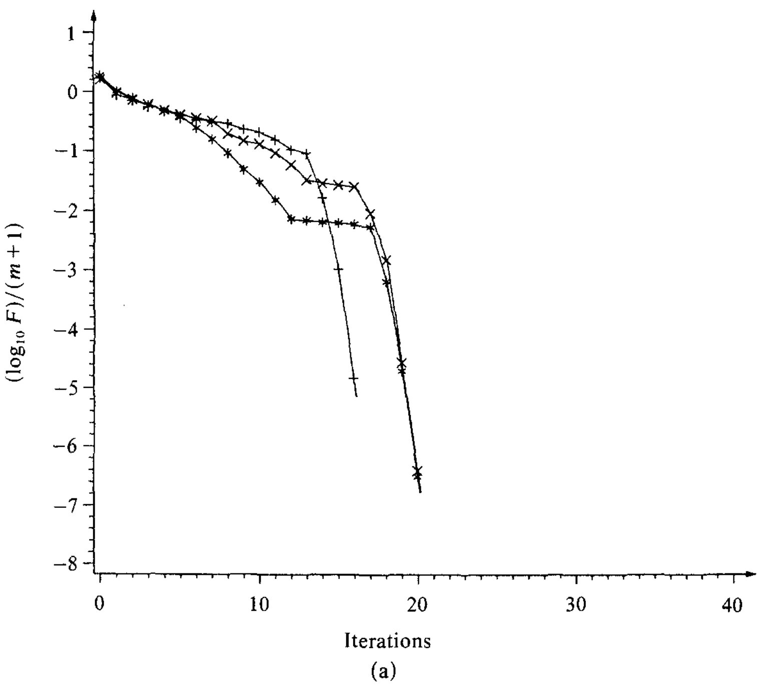
11. Assignment Problems. Example ⟨A⟩: Assignment problems were taken up as a typical class of test problems which are sparse and structured, although they might be too special in structure, being highly degenerate. The problems are of the following form:

$$(11.1)\qquad \begin{aligned} &\max \sum_{i=1}^{k}\sum_{j=1}^{k} c_{ij}x_{ij},\\ &\sum_{i=1}^{k} x_{ij}\le 1,\\ &\sum_{j=1}^{k} x_{ij}\le 1,\\ &x_{ij}\ge 0, \end{aligned}$$

where $c_{ij}(i, j=1, \ldots, k)$ are random real numbers in the interval $(0, 1)$. The number N of variables of the original problems is k^2, and the size parameters m, n of the extended problem (7.2), (7.3) are $m=2(k+1)^2$ and $n=(k+1)^2$. Computational experiments were performed for the cases with $k=2, 4, 6, 8$. The results are shown in Figures 6 and 7.

As is seen from Figure 6, the number of iterations required by the algorithm for the assignment problems is remarkably few.

12. Computational Results on the Linear Subiteration Scheme. This section presents computational results on the linear subiteration scheme given in Section 6.2. We repeat here the subiteration process as long as $\log_{10} F(x)$ decreases at least 0.1. The subiteration scheme was tested for the three classes of the above-mentioned structured problems, ⟨KM⟩, ⟨R⟩, and ⟨A⟩. The computational results



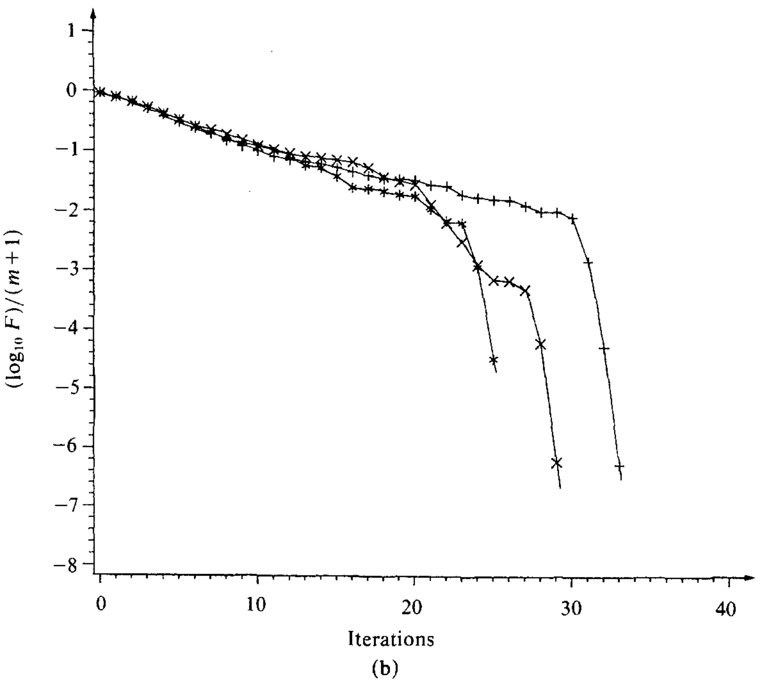


Fig. 5. Computational results for Example ⟨R⟩: the decrease of the barrier function. Three random problems with (a) $M = N = 20$ and (b) $M = N = 50$.

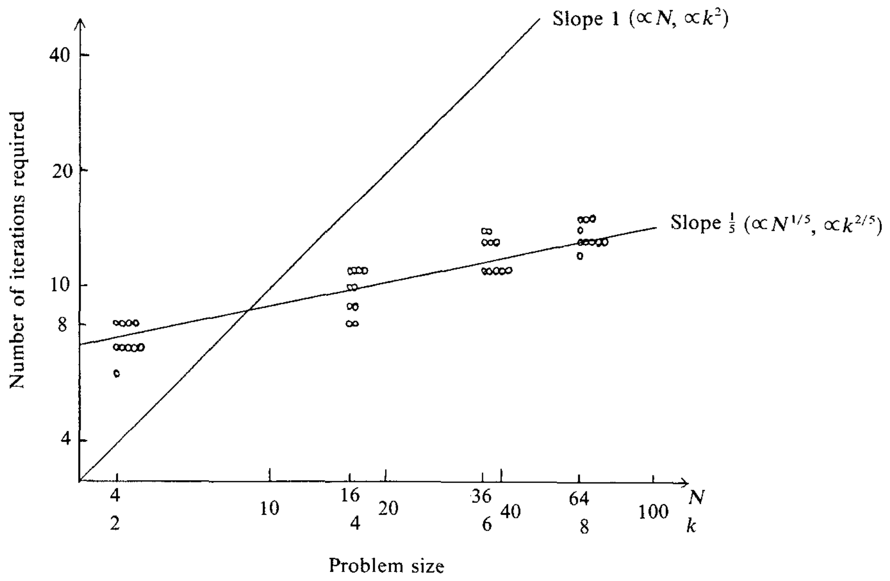


Fig. 6. Computational results for Example ⟨A⟩ (assignment problems): the number of iterations required.

are shown in Figures 8–10. In these figures, the "number of iterations" denotes the number of LU-decompositions of the Hessian matrices and not the number of subiterations in total (note that a subiteration is less costly compared with an LU-decomposition). The average number of successful subiterations per iteration was 1.4, 1.2, and 1.1 for ⟨KM⟩, ⟨R⟩, and ⟨A⟩, respectively.

From these results, it is observed that the linear subiteration scheme effiectively reduces the total number of iterations (the number of LU-decompositions) in general. For the assignment problems ⟨A⟩, the subiteration scheme reduces the total number of iterations to 70% on average. For the random problems ⟨R⟩ especially, even the speed of increase of the number of iterations with the problem size seems substantially reduced.

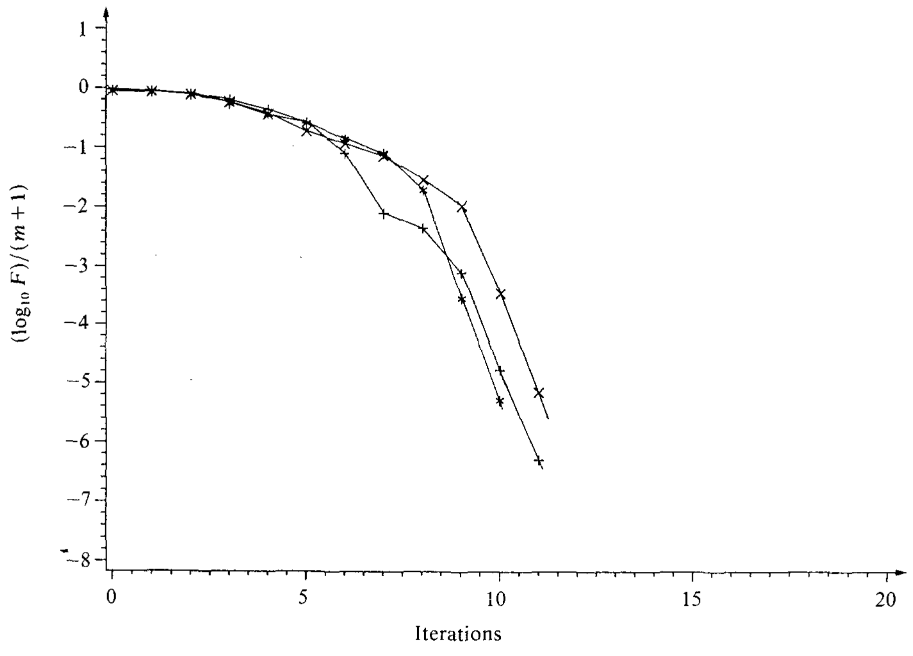
13. Conclusions to Part II.

From the preliminary computational experiments of the preceding sections the following two points may be concluded:

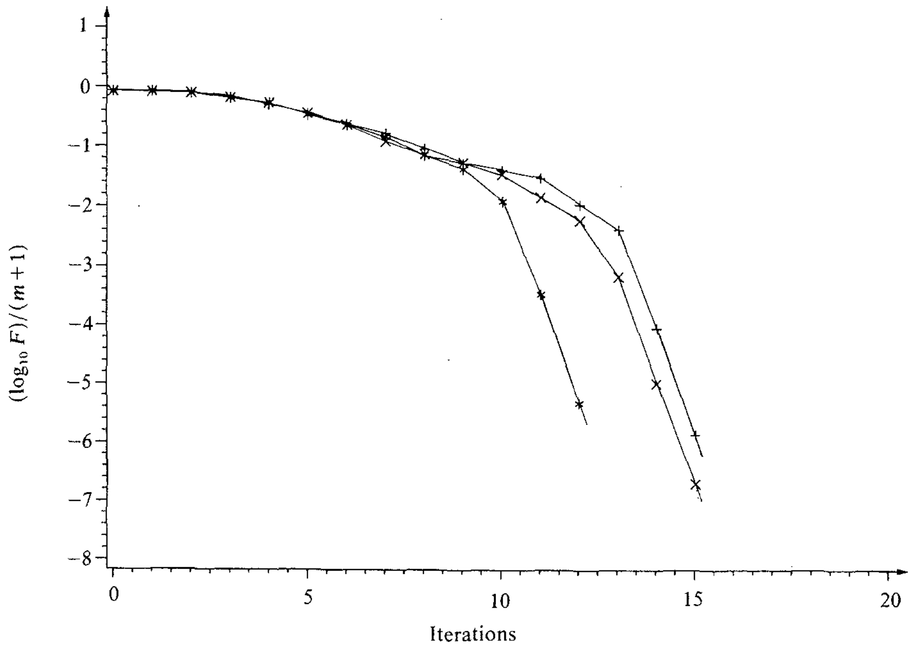
(i) The global linear convergence and the local superlinear convergence at the final stages of the algorithm proposed in Part I are experimentally confirmed.

(ii) The number of iterations required by the algorithm is $O(n^{\alpha})$ with α a little less than 1, and, for structured problems, it happens to be $O(n^{\varepsilon})$ with a small $\varepsilon < 1$.

This suggests that the algorithm becomes comparable and even faster than the simplex method if an efficient way of computing $\xi = -H^{-1}\eta$ is devised.



(a)



(b)

Fig. 7. Computational results for Example ⟨A⟩: the decrease of the barrier function. Three random problems with (a) $k=4$ and (b) $k=8$.

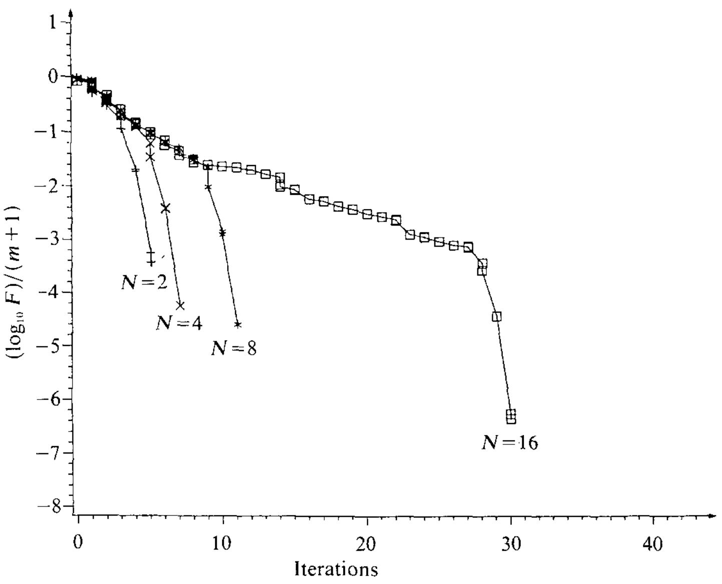


Fig. 8. Computational results of the subiteration scheme for Example ⟨KM⟩.

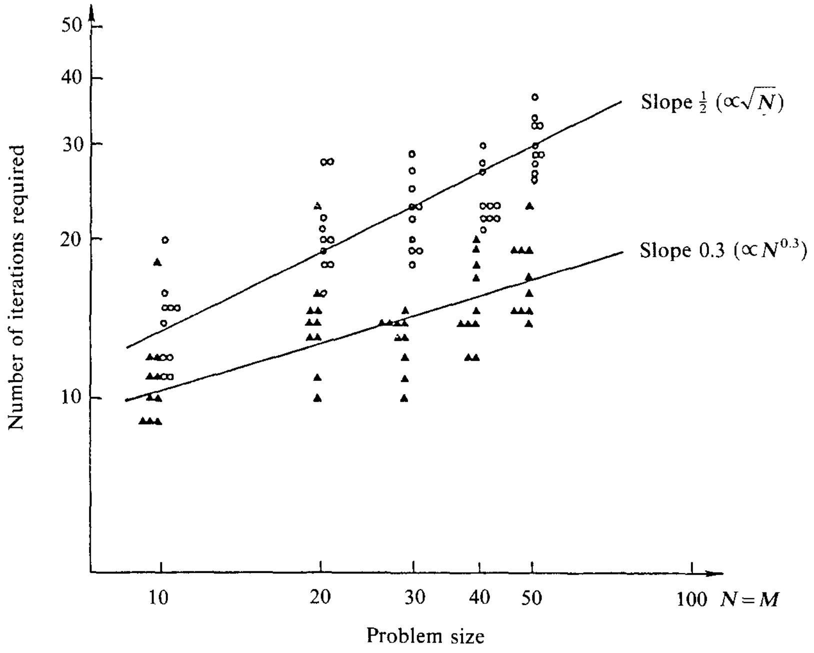


Fig. 9. Computational results concerning the subiteration scheme for Example ⟨R⟩: the number of iterations required with (○) and without (▲) the subiteration scheme.

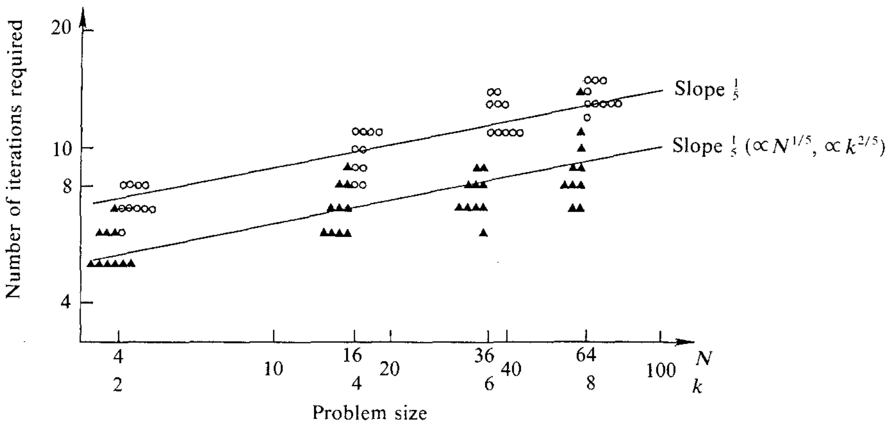


Fig. 10. Computational results concerning the subiteration scheme for Example ⟨A⟩: the number of iterations required with (▲) and without (○) the subiteration scheme.

References

[1] D. Avis and V. Chvátal, Note on Bland's pivoting rule, *Math. Programming Stud.*, **8** (1978), 24-34.

[2] M. Iri, Another "simple and fast" algorithm for linear programming, paper presented at the 12th International Symposium on Mathematical Programming, August 5-9, 1985, MIT, Boston.

[3] M. Iri and H. Imai, A method of solving linear programming - with reference to the Karmarkar method and the penalty function method, Research Meeting of the Mathematical Programming Research Group of the Operations Research Society of Japan, February 16, 1985.

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

[5] L. G. Khachian, A polynomial algorithm in linear programming, *Dokl. Akad. Nauk SSSR*, **244** (1979), 1093-1096 (in Russian); transl. in *Soviet Math. Dokl.*, **20** (1979), 191-194.

[6] L. G. Khachian, Polynomial algorithms in linear programming, *Zh. Vychisl. Mat. i Mat. Fiz.*, **20** (1980), 51-68 (in Russian); transl. in *U.S.S.R. Comput. Math. and Math. Phys.*, **20** (1980), 53-72.

[7] V. Klee and G. J. Minty, How good is the simplex algorithm?, in *Inequalities* III (O. Shisha, ed.), Academic Press, New York, 1972, pp. 159-175.