

# INTERIOR-POINT METHODS AND MODERN OPTIMIZATION CODES

Goran Lešaja

Department of Mathematics and Computer Science, Georgia Southern University, Statesboro  
E-mail: goran@gsaix2.cc.gasou.edu

---

*During the last fifteen years we have witnessed an explosive development in the area of optimization theory due to the introduction and development of interior-point methods. This development has quickly led to the development of new and more efficient optimization codes. In this paper, the basic elements of interior-point methods for linear programming will be discussed as well as extensions to convex programming, complementary problems, and semidefinite programming. Interior-point methods are polynomial and effective algorithms based on Newton's method. Since they have been introduced, the classical distinction between linear programming methods, based on the simplex algorithm, and those methods used for nonlinear programming, has largely disappeared. Also, a brief overview of some implementation issues and some modern optimization codes, based on interior-point methods, will be presented. As of now, there is no doubt that for large-scale linear programming problems these new optimization codes are very often more efficient than classical optimization codes based on the simplex method.*

**Keywords:** interior-point methods, optimization code, linear and nonlinear programming, convex programming, semidefinite programming, Newton method, Mehrota's algorithm.

---

## 1. INTRODUCTION

During the last fifteen years we have witnessed an explosive development in the area of optimization theory due to the introduction and development of interior-point methods. This development has quickly led to the development of new and more efficient optimization codes, particularly in the field of Linear Programming (LP).

The purpose of this paper is to survey the main aspects of this exciting development. The paper is organized as follows. In Section 2 we briefly review the main steps in the theoretical development of interior-point methods, mainly for LP. In Section 3 the basic idea and key elements of interior-point methods for LP are described. The extensions to some nonlinear programming problems such as linear and nonlinear complementarity problems, convex programming problems, and semidefinite programming are outlined in Section 4. Section 5 contains a brief discussion on implementation issues in the development of modern interior-point codes and a short description of some commercial and public domain linear programming codes.

## 2. BRIEF REVIEW

In this section we give a brief review of the main steps involved in the development of interior-point methods for the linear programming problem (LP) and for other optimization problems without any intention to be exhaustive.

It is not necessary to elaborate on the applicability of LP. The number of its applications in industry, business, science and other fields is enormous. That is probably the reason why advances in the theory and practice of LP get significant attention even outside the field of optimization.

Dantzig's simplex algorithm [16] for LP, developed in 1947, initiated strong research into the area of LP and optimization in general. The main idea of this algorithm is to "walk" from vertex to vertex along the edge of a feasible region (a polytope) on which the objective function is decreasing. Its popularity is due to its efficiency in solving practical problems. Years of computational experiments and applications resulted in better and better variants of this algorithm, commonly called pivoting algorithms. Computer implementations of some of these algorithms include sophisticated numerical procedures in order to achieve accuracy, stability and an ability to handle large-scale problems. Computational experience has shown that the usual number of iterations to solve the problem is  $O(n)$  where  $n$  is the number of variables in the problem. Another reason for the popularity of the simplex method and its variants is their suitability for sensitivity analysis, which is extremely important in practice. The combinatorial nature of the algorithm allowed a large number of generalizations for combinatorial problems such as the transportation problem and other network problems. Another generalization is the development of the pivoting methods for the Linear Complementarity Problem (LCP). An in-depth review of the methods for LCP can be found in [14].

Unfortunately, pivoting algorithms are not polynomial algorithms although they are finite procedures. Klee and Minty [30], in 1971, provided an LP example for which some pivoting algorithms need an exponential number of pivots. In 1978, Murty [48] provided a similar example for LCP. The good thing about these examples is that they are artificial, that is, they have not been observed in practice. This discrepancy between the worst-case complexity of pivoting algorithms and their successful practical performance, led to strong research interest in the average complexity of some pivoting algorithms in the early 80s. Adler and Megido [3] showed that for a certain probability model the number of iterations of Dantzig's self-dual parametric algorithm [16] is  $\Omega(\min\{n, m\}^2)$  where  $n$  is the number of variables and  $m$  is the number of equations. See also [56], [13], [39].

Although pivoting methods for LP and LCP have been a great success, computational experience with these methods has shown that their efficiency and numerical stability decreases as the problem dimension increases. One reason for this is the inability of these methods to preserve sparsity; thus data storage requirements increase rapidly. Another reason for this is the poor handling of round-off-errors. These unfavorable numerical characteristics together with an exponential worst case complexity (relaxed quite a bit with the artificiality of the examples in which it occurs



and in the average-case analysis) justify the need for a better, hopefully polynomial algorithm. This hope that a polynomial algorithm for LP exists was based on the fact that LP is not a NP-hard problem [39]. Finally, in 1979, more than 30 years after the appearance of the simplex algorithm, Khachiyan [29] proposed the first polynomial algorithm for LP, the so-called ellipsoid algorithm, by applying Shor's original method [55] developed for nonlinear convex programming. The publicity for this was enormous and the news even appeared in the New York Times. Similarly as for the simplex algorithm, immediate generalizations to convex quadratic programming and some classes of LCP were made. Also Grotchel et al. [19] used an ellipsoid method as a unifying concept to prove polynomial complexity results for many important combinatorial problems. Unfortunately, computational experiments soon showed that from a practical point of view the ellipsoid algorithm is useless. It performs much worse than the simplex algorithm on most practical problems and the various modifications could not offer much help. See [11] for a survey.

In late 1984, Karmarkar [31] proposed a new polynomial algorithm for LP that looked like it would perform well in practice. The main idea behind this algorithm is quite different than that of the simplex algorithm. This method is not a finite procedure, it is an iterative algorithm and the iterates are calculated not on the boundary, but in the interior of the feasible region. This algorithm makes use of projective transformations and the so-called potential function (Karmarkar's potential function). It requires  $O(nL)$  iterations where  $L$  is the size of the problem. In addition, each iteration requires  $O(n^3)$  arithmetic operations. The appearance of this algorithm started major research activity in the field of LP and other related areas, starting with the field of interior-point methods. This powerful activity has lasted ever since and the number of papers on this subject goes well into the thousands. For a while Kranich [37] maintained a detailed bibliography on interior-point methods. Recently, Wright has been maintaining a web site on interior-point methods (<http://www.mcs.anl.gov/home/otc/InteriorPoin/>) at Argonne National Laboratories which includes a list of the most recent papers and preprints in this field.

Soon a connection to the barrier and the Newton-type methods was established [20]. Renegar [54] proposed a first path-following Newton-type algorithm which further improved the complexity to  $O(\sqrt{n}L)$  number of iterations and this remains the best known worst-case complexity for LP so far. Many researchers have proposed different interior-point methods with  $O(nL)$  or even  $O(\sqrt{n}L)$  complexity. They can be put into two main groups: the potential-reduction algorithms [64] that are based on the constant reduction of some potential function at each iteration, and the path-following algorithms [35] based on approximately tracing a so-called central trajectory or central path studied first by Megiddo [45]. Actually, these two groups are not that far apart because, with a certain choice of parameters, iterates obtained by the potential-reduction algorithm stay in the so-called horn neighborhood of the central path. In each group there are algorithms based on primal, dual, or primal-dual formulation of LP. A different approach to interior-point methods is based on the concept of analytic centers and this approach was first studied by Sonnevend [57]. An in-depth review of many of the interior-point methods can be found in [66].

Throughout the history of science and mathematics, it has been the case that a new method is actually a rediscovered old method. This is exactly the case with interior-point methods. The logarithmic barrier method was actually introduced by Frisch [18] in 1955. The method of analytic centers was first suggested by Huard [23] in 1965. Also, the affine-scaling algorithm proposed by Barnes [12] and Vanderbei et al. [61] as a simplified version of Karmarkar's algorithm appeared to be just a rediscovery of a method developed by Dikin [15] in 1967. Interior-point methods were extensively studied in the 1960s and the results were best summarized in the classic monograph by Fiacco and McCormick [17] and it provided an in-depth analysis of the Sequential Unconstrained Minimization Techniques (SUMT) to solve Nonlinear Programming problems (NLP). Thus, early interior-point methods were developed for solving NP, not LP. However, these methods were soon abandoned due to the computational difficulties involved. Lootsma [41] and Murray showed [50] that the Hessian of the logarithmic barrier function, with which the system needs to be solved at each iteration, becomes increasingly ill-conditioned when the iterates approach an optimal solution. These Confidential Page 4 24.05.00 computational difficulties, coupled with the fact that for LP the simplex method performed reasonably well in practice, were the main reasons why interior-point methods were not applied on LP. If they had been, SUMT would have been shown to be a polynomial method for LP as was formally shown by Anstreicher much later on [9].

There are several reasons for the success of interior-point methods when they were rediscovered in 1985 following the appearance of Karmarkar's seminal paper [31]. First, they were immediately tried on LP and good polynomial complexity bounds were established. Second, at each iteration of an interior-point method it was necessary to usually solve the problems of a sparse linear system that becomes increasingly ill-conditioned as we approach the solution. However, the ill-conditioning in the LP case is less severe. Third, in the past two decades, hardware and software have improved so much that it is now possible to avoid ill-conditioning and solve these sparse linear systems efficiently and accurately. This is due to the advances in numerical linear algebra in general and in sparse Cholesky factorization in particular. See Andersen et al. [4] for details.

One drawback of the first interior-point methods was that they required a prior knowledge of an interior feasible point. To find such a point, or to show that it does not exist, may be just as difficult as solving the problem itself. Soon this restrictive requirement was removed and infeasible interior-point methods were proposed by Kojima et al. [32]. A particularly successful approach was one by introduced Ye et al. [70] (see also Xu et al. [63]) where a feasible primal-dual interior-point algorithm was applied to the homogeneous, self-dual, always feasible reformulation of the original LP problem. This reformulation has a slightly bigger dimension than the original problem, but this is a small price to pay for the good properties that are obtained. One such property is the one that allows for the detection of infeasibility in the original problem.

A drawback with the nice  $O(\sqrt{n}L)$  worst-case complexity result for some path-following algorithms was that it was obtained by taking short steps, i. e., steps in a small neighborhood of the central path. Computationally, that is undesirable and we



would like to be able to take much bigger steps. Lustig's et al. implementation [38] that actually employs large steps has been really successful although their implementation does not even guarantee global convergence. Some other large-step algorithms were proved to be globally convergent, however the best complexity for these algorithms stays  $O(nL)$ , and this is worse than for short-step algorithms, although they perform better in practice than the short-step algorithms. To bridge the gap Hung and Ye [25] proposed an  $r$ -order algorithm with  $O(n^{(r+1)/2r}L)$  complexity. The main idea was that instead of just calculating a search direction,  $r$  additional higher-order corrections of a search direction are calculated in each iteration, in order to find a better search direction and, thus, reduce the number of iterations. Earlier, Mehrota [42] studied the implementation of higher-order methods but did not discuss global convergence. He showed that the first correction gives the biggest improvement in the practical performance of the algorithm. These types of algorithms are called predictor-corrector algorithms and they are now dominant in practical implementations. Also, computational experiments have shown that a primal-dual formulation is superior to either a primal or dual formulation of the algorithm [44].

Besides global convergence and polynomial worst-case complexity, a desirable feature of an algorithm is its fast local convergence. Yamashita [69] was the first to prove quadratic convergence of the polynomial primal algorithm, but under the very restrictive assumptions that the optimal objective value is known and that the iteration sequence converges to a nondegenerate optimal vertex. Then, Zhang and Tapia [72] showed that for a certain choice of parameters a polynomial primal-dual algorithm converges quadratically for nondegenerate problems and superlinearly for degenerate problems under the assumption that the iteration sequence converges. Finally, Ye et al. [68] and Mehrota [43] proved that the Mizuno-Todd-Ye  $O(\sqrt{n}L)$  algorithm [49] is quadratically convergent without any assumptions.

Interior-point methods are iterative algorithms and they produce only an approximation to the solution of the problem. This approximation can be as good as desired, thus, it is called an  $\varepsilon$ -approximate solution. However, for LP problems it is possible to recover the exact (vertex) solution of a problem if the  $\varepsilon$ -approximate solution is close enough to that exact solution [65], [46]. These procedures include finite termination procedures and cross-over procedures. They transform interior-point methods for LP to theoretically finite algorithms that can produce an exact solution to a problem. For many problems in practice an  $\varepsilon$ -approximate solution is sufficient, but there are applications where an exact solution is needed which underscores the practical value of finite termination and cross-over procedures as well.

The tradition of generalization from LP to other optimization problems continued even more strongly in the case of interior-point methods. Many methods were first extended to LCP, some of them still maintaining the best-known  $O(\sqrt{n}L)$  complexity. See, for example [36], [67]. Although interior-point methods were originally developed in the 1960s [17] to solve Nonlinear Programming problems (NLP), recent in-depth analysis of interior-point methods for LP opened new research directions in the study of interior-point methods for NLP. In their seminal monograph, Nesterov and Nemirovskii [51] provided a unified theory of polynomial interior-point methods for a

large class of convex programming problems that satisfies the so-called self-concordancy condition. Significant advances have also been made in interior-point methods for the Nonlinear Complementarity Problem (NCP) [10]. Recently, the development of interior-point methods for Semidefinite Programming (SDP) has been a very active research area. This activity is partially due to the fact that many important problems in combinatorics, control theory, pattern recognition, etc., can be formulated as SDP. See for example [6].

### 3. INTERIOR-POINT METHODS FOR LP

In this section we present a generic infeasible interior-point algorithm for the LP problem. Consider an LP problem in its standard form: Given the data  $b \in R^m$ ,  $c \in R^n$ ,  $A \in R^{m \times n}$  find  $x \in R^n$  that solves the problem

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0. \end{aligned} \tag{3.1}$$

The vector  $x \in R^n$  is called a vector of primal variables and the set  $\Phi_p = \{x : Ax = b, x \geq 0\}$

is called a feasible region.

The corresponding dual problem is then given by

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & A^T y + s = c, \\ & s \geq 0. \end{aligned} \tag{3.2}$$

The vector  $y \in R^m$  is called a vector of dual variables and the vector  $s \in R^m$  is called a vector of dual slack variables. Also, denote the dual feasible region as  $\Phi_d$ .

There is a well known theory that relates primal and dual LP problems and their solutions with weak and strong duality theorems being at its core. It will not be elaborated on here since it can be found in any standard textbook on LP. See for example [16].

Consider now a logarithmic barrier reformulation for primal problem (3.1).

$$\begin{aligned} \min \quad & c^T x - \mu \sum_{i=1}^n \ln x_i \\ \text{s.t.} \quad & Ax = b, \\ & x > 0. \end{aligned} \tag{3.3}$$

Problems (3.1) and (3.3) are equivalent in the sense that they have the same solution sets. The Karush-Kuhn-Tucker (KKT) conditions for problem (3.3) are derived from the Lagrange function of this problem

$$L(x, y) = c^T x - \mu \sum_{i=1}^n \ln x_i - y^T (Ax - b), \tag{3.4}$$



and they are

$$\begin{aligned} \nabla_x L(x, y) &= c - \mu X^{-1} e - A^T y = 0, \\ \nabla_y L(x, y) &= b - Ax = 0, \\ x &> 0, \end{aligned} \tag{3.5}$$

where  $X \in R^{n \times n}$  represents a diagonal matrix with the components of the vector  $x \in R^n$  on its diagonal,  $e \in R^n$  is a vector of ones, and  $\mu \in R$  is a parameter. Using the transformation  $s = \mu X^{-1} e$  system (3.5) becomes

$$\begin{aligned} A^T y + s &= c, \\ Ax = b, x &> 0, \\ Xs &= \mu e. \end{aligned} \tag{3.6}$$

The logarithmic barrier model for the dual LP problem (3.2) is

$$\begin{aligned} \max \quad & b^T y + \mu \sum_{i=1}^n \ln s_i \\ \text{s.t.} \quad & A^T y + s = c, \\ & s > 0. \end{aligned} \tag{3.7}$$

The KKT conditions for the above problem are

$$\begin{aligned} \nabla_x L(x, y, s) &= A^T y + s - c = 0, \\ \nabla_y L(x, y, s) &= b - Ax = 0 = 0, \\ \nabla_s L(x, y, s) &= \mu s^{-1} e - x = 0, \\ s &> 0, \end{aligned} \tag{3.8}$$

or equivalently

$$\begin{aligned} A^T y + s - c &= 0, s > 0, \\ b - Ax &= 0, \\ Xs &= \mu e. \end{aligned} \tag{3.9}$$

Combining KKT conditions for the primal (3.6) and dual (3.9) barrier models we obtain the primal-dual KKT conditions

$$\begin{aligned} A^T y + s - c &= 0, s > 0, \\ b - Ax &= 0, x > 0, \\ Xs &= \mu e. \end{aligned} \tag{3.10}$$

The above conditions are very similar to the original KKT conditions for LP.

$$\begin{aligned} A^T y + s - c &= 0, s \geq 0, && \leftarrow \text{Dual feasibility} \\ b - Ax &= 0, x \geq 0 && \leftarrow \text{Primal feasibility} \\ Xs &= 0. && \leftarrow \text{Complementarity} \end{aligned} \tag{3.11}$$

The only differences between (3.10) and (3.11) are strict positivity of the variables and perturbation of the complementarity equation. Although these differences seems minor, they are essential in devising a globally convergent interior-point algorithm for LP.

Note that the complementarity equation in (3.11) can be written as  $x^T s = 0$ . It is a well known fact that  $x^T s = b^T y - c^T x$  and therefore  $x^T s$  can be viewed as a primal-dual gap between objective functions. Hence, the complementarity condition in (3.11) can be interpreted as the condition of primal-dual gap being zero, which is just another look at the strong duality theorem for LP.

It is a well known fact that  $(x^*, y^*, s^*)$  is a solution to the problem (3.11) iff  $x^*$  is a solution of the primal LP problem (3.1) and  $(y^*, s^*)$  is a solution of the dual LP problem (3.2). Thus, any barrier method for solving a problem (3.11) consists of the solving of a sequence of systems (3.10) with different values  $\mu_k$  for the parameter  $\mu$ . If a strictly monotone decreasing sequence  $\{\mu_k\}$  exists, such that  $\lim_{k \rightarrow \infty} \mu_k = 0$ , then any accumulation point of the solution sequence  $\{(x^k, y^k, s^k)\}$  is a solution of the original primal LP problem (3.1) and associated dual problem (3.2). More formally, the generic barrier method can be stated as follows.

- (BM)** 1. Given  $\mu_k$  solve THE system (3.10).  
 2. Decrease  $\mu_k$  to  $\mu_{k+1}$ .  
 3. Set  $k \leftarrow k + 1$  and go to 1.

The standard method used for solving the system (3.10) in Step 1 in the above algorithm is one step of the modified (dumped) Newton method while the standard choice for  $\mu_k$  is

$$\mu_k = \frac{(x^k)^T s^k}{n}. \quad (3.12)$$

Note that the application of one step of the modified Newton method will provide us only with an approximate solution to the system (3.10). However, it can be shown that this is sufficient for the sequence of iterates to converge to the solution.

One step of the modified Newton method is formalized below.

- (MNM)** 1. Given an iterate  $x^k$ , find the search direction  $d_x$  by solving the linear system  $\nabla f(x^k) d_x = -f(x^k)$ .  
 2. Find step size  $\alpha_k$ .  
 3. Update  $x^k$  to  $x^{k+1} = x^k + \alpha_k d_x$ .

The symbol  $\nabla f$  represents derivative, gradient, or the Jacobian of the function  $f$  depending on the definition of the function  $f$ . The choice of  $\alpha_k$  in the original Newton method is  $\alpha_k = 1$ . In the modified Newton method the choice of  $\alpha_k$  may be

$$\alpha_k = \arg \min_{\alpha \in [0,1]} \|f(x^k + \alpha d_x)\|.$$

The good thing about this choice is that we get  $\|f(x^{k+1})\| \leq \|f(x^k)\|$ , that is,  $d_x$  is a decreasing direction.

An application of the MNM algorithm to the system (3.10) can be viewed as an application of MNM to the function



$$F_\gamma(x, y, s) = \begin{bmatrix} Ax - b \\ A^T y + s - c \\ Xs - \gamma\mu e \end{bmatrix} \quad (3.13)$$

Note that the original system (3.10) has been slightly modified by adding the scaling factor  $\gamma$  to the last equation, with the intention of increasing the flexibility of the algorithm. Thus, a search direction is a solution of the system

$$\nabla F_\gamma(x^k, y^k, s^k) \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = -F_\gamma(x^k, y^k, s^k), \quad (3.14)$$

or equivalently

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = \begin{bmatrix} b - Ax^k \\ c - s^k - A^T y^k \\ -X^k s^k + \gamma\mu_k e \end{bmatrix} = \begin{bmatrix} r_P^k \\ r_D^k \\ -X^k s^k + \gamma\mu_k e \end{bmatrix}. \quad (3.15)$$

At this point we would like to comment on why the perturbed KKT conditions for LP given by the system (3.10) were used instead of the original KKT conditions given by the system (3.11). Suppose we apply the Newton method directly to the system (3.11). Particularly, an application of the Newton method to the last (complementarity) equation leads to

$$Sd_x + Xd_s = -Xs,$$

or equivalently

$$s_i(d_x)_i + x_i(d_s)_i = -x_i s_i, \quad \forall i = 1, \dots, n.$$

If  $x_i = 0$  and  $s_i > 0$  for some index  $i$ , then an immediate consequence of the above equation is  $(d_x)_i = 0$  and the update  $x_i + \alpha(d_x)_i$  becomes 0, and stays 0 forever. The iteration sequence may get “stuck” at the wrong face of  $R_+^n$  and never converge to the solution. To avoid this “trapping” phenomenon we perturb the complementary equation

$$Xs = \mu e, \mu > 0.$$

The step size is connected to the concept of the central path

$$\Gamma = \{(x, s) : (x, s) \text{ feasible}, Xs = \mu e, \mu > 0\}, \quad (3.16)$$

that is, a trajectory parametrized by the parameter  $\mu$ . It has been shown by Megiddo [45] that this trajectory leads to a strictly complementary solution when  $\mu \rightarrow 0$ . A strictly complementary solution is defined as a pair of solutions  $x^*$  and  $s^*$  such that  $x^* + s^* > 0$ . Goldman and Tucker showed [21] that such a solution always exists for LP if the primal and dual problems are both feasible. Moreover, Guler and Ye [22] showed that the supports

$$P^* = \{j : x_j > 0\} \text{ and } Z^* = \{j : s_j > 0\}$$

are invariant for all pairs of strictly complementary solutions. The neighborhoods (horn neighborhoods) of the central path can be defined using different norms

$$N_2(\beta) = \{(x, s) : \|Xs - \mu e\|_2 \leq \beta\mu\} \quad (3.17)$$

$$N_\infty(\beta) = \{(x, s) : \|Xs - \mu e\|_\infty \leq \beta\mu\} \quad (3.18)$$

or even a pseudonorm

$$N_\infty^-(\beta) = \{(x, s) : \|Xs - \mu e\|_\infty^- \leq \beta\mu\} = \{(x, s) : Xs \geq (1 - \beta)\mu\} \quad (3.19)$$

where  $\|z\|_\infty^- = \|z^-\|_\infty$  and  $(z^-)_j = \min\{z_j, 0\}$ . These neighborhoods are related as follows:

$$\Gamma \subseteq N_2(\beta) \subseteq N_\infty(\beta) \subseteq N_\infty^-(\beta). \quad (3.20)$$

Hence, if a step size is chosen in such a way that the iterates stay in the one of the above horn neighborhoods we are guaranteed a convergence of the method. Let

$$x(\alpha) = x^k + \alpha d_x, \quad s(\alpha) = s^k + \alpha d_s.$$

The step size  $\alpha$  is chosen in such a way that  $(x(\alpha), s(\alpha))$  belongs to one of the above neighborhoods, i. e.,

$$\alpha_k = \max\{\alpha' : \|X(\alpha)s(\alpha) - \mu(\alpha)e\| \leq \beta\mu(\alpha), \forall \alpha \in [0, \alpha']\} \quad (3.21)$$

where

$$\mu(\alpha) = \frac{x^T(\alpha)s(\alpha)}{n}. \quad (3.22)$$

Although the Newton method is not necessarily globally convergent, by using the above technique, global convergence is guaranteed. Moreover, fast local convergence (quadratic or at least superlinear) is preserved. Now, the first step of the barrier algorithm BM can be completed by calculating the new iterates

$$x^{k+1} = x^k + \alpha_k d_x, \quad s^{k+1} = s^k + \alpha_k d_s. \quad (3.23)$$

The second step of BM is the calculation of  $\mu_{k+1}$  using equation (3.22). It can be shown that  $\mu_{k+1} < \mu_k$  as it is required. Finally, note that BM can produce only an approximate solution. An iterate  $(x^k, y^k, s^k)$  is an  $\varepsilon$ -approximate solution if

$$\|Ax^k - b\| \leq \varepsilon_p, \quad \|A^T y^k + s^k - c\| \leq \varepsilon_D, \quad (x^k)^T s^k \leq \varepsilon_G \quad (3.24)$$

for a given  $(\varepsilon_p, \varepsilon_D, \varepsilon_G) > 0$ .

The interior-point algorithm can now be summarized as follows:

### Algorithm (IPM)

#### Initialization:

1. Choose  $\beta, \gamma \in (0, 1)$  and  $(\varepsilon_p, \varepsilon_D, \varepsilon_G) > 0$ . Choose  $(x^0, y^0, s^0)$  such that

$$(x^0, s^0) > 0 \text{ and } \|X^0 s^0 - \mu_0 e\| \leq \beta\mu_0 \text{ where } \mu_0 = \frac{(x^0)^T s^0}{n}.$$

2. Set  $k = 0$ .



**Step:**

3. Set  $r_p^k = b - Ax^k, r_D^k = c - A^T y^k - s^k, \mu_k = \frac{(x^k)^T s^k}{n}$ .

4. Check the termination. If

$$\|r_p^k\| \leq \varepsilon_p, \|r_D^k\| \leq \varepsilon_D, (x^k)^T s^k \leq \varepsilon_G,$$

then terminate.

5. Compute the direction by solving a system

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_D^k \\ -X^k s^k + \gamma \mu_k e \end{bmatrix}. \quad (3.25)$$

6. Compute the step size

$$\alpha_k = \max\{\alpha' : \|X(\alpha)s(\alpha) - \mu(\alpha)e\| \leq \beta\mu(\alpha), \forall \alpha \in [0, \alpha']\},$$

where  $x(\alpha) = x^k + \alpha d_x, s(\alpha) = s^k + \alpha d_s, \mu(\alpha) = \frac{x^T(\alpha)s(\alpha)}{n}$ .

7. Update

$$\begin{aligned} x^{k+1} &= x^k + \alpha_k d_x, \\ y^{k+1} &= y^k + \alpha_k d_y, \\ s^{k+1} &= s^k + \alpha_k d_s. \end{aligned}$$

8. Set  $k = k + 1$  and go to step 3.

The above algorithm has favorable convergence properties. For a certain choice of the parameters and using the neighborhood  $N_2(\beta)$ , the following convergence results can be obtained.

- **Global convergence:** The algorithm IPM will achieve an  $\varepsilon$ -approximate solution in  $O(\sqrt{n} \log 1/\varepsilon)$  iterations, where  $\varepsilon = \min\{\varepsilon_p, \varepsilon_D, \varepsilon_G\}$ .
- **Local convergence:** For a sufficiently large  $k$  there exists a constant  $\lambda > 0$  such that  $x_i^{k+1} s_i^{k+1} \leq \lambda(x_i^k s_i^k)^2, \forall i = 1, \dots, n$ .

There are many modifications and variations of this algorithm. In fact this algorithm represents a whole class of algorithms. We can consider different neighborhoods of a central path. Because of the relation (3.20), if  $N_2(\beta)$  is used, the algorithm IPM is called a short-step algorithm, and if  $N_\infty(\beta)$  or  $N_\infty^-(\beta)$  is selected, the algorithm IPM is called a long-step algorithm. Unfortunately, the price to pay for taking bigger steps in a long-step algorithm is worse global convergence, that is, algorithm needs  $O(n \log 1/\varepsilon)$  to achieve an  $\varepsilon$ -approximate solution. Details of the IPM algorithm and the proofs of the convergence results can be found in [66].

Note that in the algorithm IPM only one step of the modified Newton method MNM was used to find an approximate solution of system (3.10). However, more steps of the MNM method can be performed in each iteration in order to achieve better approximation. The IPM algorithm is then called a higher-order algorithm. If only one

additional step per iteration is performed, the algorithm is called a predictor-corrector algorithm.

### Predictor-Corrector Algorithm (PC-IPM)

#### Initialization:

1. Choose  $\beta_2, \beta_1, \in (0,1)$  and  $(\varepsilon_p, \varepsilon_D, \varepsilon_G) > 0$ . Let  $\beta_2 < \beta_1$ . Choose  $(x^0, y^0, s^0)$  such that  $(x^0, s^0) > 0$  and  $\|X^0 s^0 - \mu_0 e\| \leq \beta_2 \mu_0$  where  $\mu_0 = \frac{(x^0)^T s^0}{n}$ .
2. Set  $k = 0$ .

#### Predictor Step:

3. Set  $r_p^k = b - Ax^k, r_D^k = c - A^T y^k - s^k, \mu_k = \frac{(x^k)^T s^k}{n}$ .
4. Check the termination. If

$$\|r_p^k\| \leq \varepsilon_p, \|r_D^k\| \leq \varepsilon_D, (x^k)^T s^k \leq \varepsilon_G,$$

then terminate.

5. Compute the direction by solving system (3.25) with  $\gamma = 0$

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = \begin{bmatrix} r_p^k \\ r_D^k \\ -X^k s^k \end{bmatrix}. \quad (3.26)$$

6. Compute the step size

$$\bar{\alpha} = \max\{\alpha' : \|X(\alpha)s(\alpha) - \mu(\alpha)e\| \leq \beta_1 \mu(\alpha), \forall \alpha \in [0, \alpha']\}.$$

7. Update

$$\bar{x} = x^k + \bar{\alpha} d_x,$$

$$\bar{y} = y^k + \bar{\alpha} d_y,$$

$$\bar{s} = s^k + \bar{\alpha} d_s.$$

#### Corrector Step:

8. Set

$$\bar{r}_p = b - A\bar{x}, \bar{r}_D = c - A^T \bar{y} - \bar{s}, \bar{\mu} = \frac{(\bar{x})^T \bar{s}}{n}.$$

9. Compute the direction by solving system (3.25) with  $\gamma = 1$

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ \bar{S} & 0 & \bar{X} \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = \begin{bmatrix} \bar{r}_p \\ \bar{r}_D \\ -\bar{X} \bar{s} + \bar{\mu} e \end{bmatrix}. \quad (3.27)$$

10. Update

$$x^{k+1} = x^k + d_x,$$

$$y^{k+1} = y^k + d_y,$$

$$s^{k+1} = s^k + d_s.$$

11. Set  $k = k + 1$  and go to step 3.



Note that in the predictor step the direction is calculated in a larger neighborhood while in the corrector step direction with a unit step size is calculated to return an iterate to a smaller neighborhood. Thus, the predictor-corrector algorithm PC-IPM can be viewed as an algorithm that combines large and short steps. Quite suprisingly, global convergence remains  $O(\sqrt{n} \log 1/\epsilon)$  and fast local convergence is preserved. In addition, algorithms of this type show the best practical performance and are therefore implemented in almost all modern interior-point codes. See [4], [42].

Note that both of the above algorithms (IPM and PC-IPM) are infeasible algorithms, that is, a starting point is not required to be feasible. First interior-point algorithms required the starting point to be feasible as well as all the subsequent iterates. In this case, system (3.25) has to be modified to

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X^k s^k + \gamma \mu_k e \end{bmatrix}. \tag{3.28}$$

The algorithms IPM and PC-IPM are also path-following algorithms since the iterates are required to stay in the horn neighborhood of the central path ((3.17)-(3.19)). These algorithms are designed to reduce primal-dual gap ( $\mu$ ) directly in each iteration. There is another group of interior-point algorithms that are designed to reduce primal-dual gap ( $\mu$ ) indirectly in each iteration. What these algorithms reduce directly is a certain potential function that is reduced for a constant in each iteration. That is why they are called potential-reduction algorithms. The iterates of these algorithms do not necessarily stay in the horn neighborhood of the central path. In this section the generic potential-reduction algorithm will not be discussed in detail. They can be found in [66]. We will only mention the most popular potential function, the so-called Tanabe-Todd-Ye primal-dual potential function

$$\Phi_\rho(x, s) = \rho \log x^T s - \sum_{i=1}^n \log x_i s_i, \tag{3.29}$$

where  $\rho > n$ . Using this function, Ye [64] developed the potential-reduction algorithm with  $O(\sqrt{n} \log 1/\epsilon)$  complexity, matching the best result obtained for path-following algorithms. Actually, Karmarkar's algorithm is also a variant of the potential-reduction algorithm with the so-called primal potential function

$$\Phi_\rho(x) = \rho \log(c^T x - Z) - \sum_{i=1}^n \log x_i, \tag{3.30}$$

where  $\rho = n + 1$  and  $Z$  is a lower bound on the optimal objective value.

#### 4. EXTENSIONS

In this section we discuss the extensions of interior-point algorithms to some classes of NLP problems such as the linear complementarity problem, the nonlinear complementarity problem, convex programming, and semidefinite programming. The

algorithms are not stated in detail, only the key elements, mainly the calculation of the search direction, are discussed.

### Linear complementarity problem

The Linear Complementarity problem (LCP) is the problem of finding a pair of vectors  $x \in R^n$ ,  $s \in R^n$  that satisfy the following condition:

$$s = Mx + q, \quad x^T s = 0, \quad (x, s) \geq 0, \quad (4.1)$$

where  $M \in R^{n \times n}$  is a given matrix and  $q \in R^n$  is a given vector.

Different classes of LCP can be obtained by imposing different additional conditions on the matrix  $M$ . The class most often analyzed and used in practice is a class of monotone LCP where the matrix  $M$  is required to be positive semidefinite. Other classes include the  $P$ -matrices (matrices with all the principal minors positive), the  $P^*$ -matrices (sufficient matrices), and the  $P_0$ -matrices (matrices with all the principal minors nonnegative). There are also other formulations of LCP besides problem (4.1), and this is called the standard LCP. A popular formulation is the horizontal LCP

$$Mx + Ns = q, \quad x^T s = 0, \quad (x, s) \geq 0. \quad (4.2)$$

For an extensive analysis of LCP see [14], [34].

Although LCP is not an optimization problem, it is closely related to optimization problems because the KKT conditions for many optimization problems can be formulated as LCP. For example, it is obvious that the KKT conditions (3.11) for LP can be viewed as standard LCP with

$$M = \begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix}, \quad q = \begin{bmatrix} c \\ -b \end{bmatrix}. \quad (4.3)$$

A similar result can be obtained for the Quadratic Programming problem (QP). In addition to serving as a unifying framework for the analysis of many optimization problems, LCP also appears as a direct formulation of many practical problems in engineering and other areas. See for example [2] and [14].

It is not difficult to formulate an interior-point method for LCP. The structure of the algorithm IPM for LP from the previous section remains the same. However, one step of the modified Newton method MNM is now applied to the different function

$$F(x, s) = \begin{bmatrix} Mx - s + q \\ Xs \end{bmatrix}. \quad (4.4)$$

Therefore, the search direction will be the solution of the system

$$\begin{bmatrix} M & -I \\ S^k & X^k \end{bmatrix} \begin{bmatrix} d_x \\ d_s \end{bmatrix} = \begin{bmatrix} r^k \\ X^k s^k + \gamma \mu_k e \end{bmatrix}, \quad (4.5)$$

where  $r^k = s^k - Mx^k - q$ . We will skip other details of this algorithm; they can be found in [66].



The analysis and the results of the interior-point methods for LP can be repeated almost completely for a monotone LCP and even for some larger classes of LCP such as the LCP with sufficient matrices. See, for example, [34], [67], and [40].

### The convex programming problem

A general Convex Programming problem (CP) can be formulated as finding a vector  $x \in R^n$  that solves the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0, \end{aligned} \tag{4.6}$$

where  $f: R^n \rightarrow R$  and  $g: R^n \rightarrow R^m$  are smooth convex functions.

There is no need to elaborate the applicability of CP since there are numerous applications in industry, finance and other areas.

The Lagrange function for this problem is

$$L(x, y) = f(x) + y^T g(x), \tag{4.7}$$

where  $y \in R^m$  is a vector of the Lagrange multipliers. Thus, the KKT conditions for the problem (4.6) are

$$\begin{aligned} \nabla_x L(x, y) &= \nabla f(x) + \nabla g(x)^T y = 0, \\ g(x) + s &= 0, \\ y^T s &= 0, \\ (y, s) &\geq 0. \end{aligned} \tag{4.8}$$

As with LP, the above KKT conditions (4.8) can be used to characterize solutions of the original CP problem (4.6), but only if an additional assumption, known as the constraint qualification, is satisfied at the solution of the system (4.8). A good description of several types of constraint qualifications can be found in [47].

Most algorithms for CP search for a point that satisfies the KKT conditions (4.8) in the hope that the constraint qualification will hold at that point. It is not difficult to generalize the algorithm IPM for the LP, described in the previous section, to this more general setting. The function to which we apply one step of the modified Newton method MNM is clearly visible in the system (4.8). The search direction is then obtained as a solution of the system

$$\begin{bmatrix} \nabla_{xx} L(x, y) & \nabla g(x)^T & 0 \\ \nabla g(x) & 0 & I \\ 0 & S & Y \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = \begin{bmatrix} -(\nabla f(x) + \nabla g(x)^T y) \\ -(g(x) + s) \\ -Ys + \gamma \mu e \end{bmatrix}, \tag{4.9}$$

where  $\mu = \frac{y^T s}{m}$ . For the sake of simplicity, the index  $k$  in the iterate  $(x^k, y^k, s^k)$  that is used in the system (4.9) was omitted. The other details of the algorithm were also omitted and can be found in [51], [28].

The convergence analysis of interior-point algorithms for CP is much harder than the one for LP because system (4.9) is much harder to analyze for that purpose. The polynomial global convergence for the general class of CP problems cannot be established [51]. In order to achieve it, additional assumptions on the smoothness of the functions are necessary. The most general smoothness condition is the self-concordant condition of Nesterov and Nemirovskii [51] which basically requires that the third derivative of the barrier function along any direction is assumed to be bounded in terms of its second derivative at all strictly feasible points. Other conditions include the relative Lipschitz condition of Jarre [26] and scaled Lipschitz condition of Potra and Ye [53]. An in-depth analysis of interior-point methods for CP can be found in the seminal monograph of Nesterov and Nemirovskii [51].

### The nonlinear complementarity problem

The standard LCP given by (4.1) can be generalized to nonlinear setting by replacing the linear function  $Mx + q$  with the smooth nonlinear function  $f : R^m \rightarrow R^m$ . Thus, the Nonlinear Complementary Problem (NCP) can be stated as the problem of finding a pair of vectors  $x \in R^n$ ,  $s \in R^n$  that satisfy the following conditions

$$s = f(x), \quad x^T s = 0, \quad (x, s) \geq 0 \quad (4.10)$$

The analogue of monotone LCP with positive semidefinite matrix  $M$  is the monotone NLP with a monotone function. A function  $f : R^m \rightarrow R^m$  is monotone if

$$(\forall x^1, x^2 \in R^n) \left( (x^1 - x^2)^T (f(x^1) - f(x^2)) \geq 0 \right), \quad (4.11)$$

for all  $x^1, x^2$  in the neighborhood of the positive orthant  $R_+^n$ . Some other classes of LCP have their generalizations to NCP as well. See [33], [40].

The importance of NCP arose partially from the obvious fact that the KKT conditions (4.8) for CP can be formulated as monotone NCP. Also, the monotone variational inequality problem that has important practical applications, can be formulated as monotone NCP. The monotone variational inequality problem can be stated as a problem of finding a vector  $x^* \in R^n$  such that

$$x^* \in \Omega, \quad (x - x^*)^T \Phi(x^*) \geq 0 \quad \forall x \in \Omega, \quad (4.12)$$

where  $\Omega \subseteq R^n$  is a closed convex set and  $\Phi : R^n \rightarrow R^n$  is a monotone function.

The interior-point algorithm for NCP can be adapted from the one for LCP in a straightforward way. The search direction is a solution of the system

$$\begin{bmatrix} \nabla f(x^k) & -I \\ S^k & X^k \end{bmatrix} \begin{bmatrix} d_x \\ d_s \end{bmatrix} = \begin{bmatrix} s^k - f(x^k) \\ -X^k s^k + \gamma \mu_k e \end{bmatrix} = \begin{bmatrix} r^k \\ -X^k s^k + \gamma \mu_k e \end{bmatrix}, \quad (4.13)$$

where

$$\mu_k = \frac{(x^k)^T s^k}{n}.$$

The other details of this algorithm are omitted and can be found in [10].



Similarly, as in the case of CP, in order to prove the polynomial convergence results an additional smoothness condition has to be imposed on the function  $f$ . Most papers analyze monotone NCP [10], [53], and [58]. There are few that deal with the larger class of sufficient NCP [27] or [40] and even fewer that deal with the local convergence [58] or [40].

### Semidefinite programming

Interior-point methods for Semidefinite Programming (SDP) have recently been a very active research area. SDP is an extension of LP in which symmetric matrices and real vectors are included among the variables, and positive semidefiniteness conditions on the matrix variables are included in the constraints.

To define the standard form of SDP and its dual it is necessary to introduce some notation. Let  $\Lambda^n$  be the set of real symmetric  $n \times n$  matrices, and define an inner product

$$G \bullet H = \text{trace}(GH) = \sum_{i,j} G_{ij} H_{ij} . \quad (4.14)$$

If  $G \in \Lambda^n$ , we use  $G \geq 0$  to denote positive semidefiniteness and  $G > 0$  to denote positive definiteness. Using this notation, we define the SDP as the problem of finding a matrix  $X \in \Lambda^n$  that solves the problem

$$\begin{aligned} \min \quad & C \bullet X \\ \text{s.t.} \quad & A(X) = b, \\ & X \geq 0, \end{aligned} \quad (4.15)$$

where  $A(X) = [A_1 X, \dots, A_m X]^T$ , with  $C \in \Lambda^n$ ,  $A_k \in \Lambda^n$  for all  $k$ , and  $b \in R^m$ . The dual of problem (4.14) is

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & A^T(y) + S = C, \\ & S \geq 0, \end{aligned} \quad (4.16)$$

where  $A^T(y) = \sum_{k=1}^m y_k A_k$  and  $S \in \Lambda^n$  is a dual slack matrix. Note that the matrices  $X, S \in \Lambda^n$  are not necessarily diagonal matrices as they were earlier in this section, and in the previous section.

Although this is a nonlinear problem, the analogy with LP is apparent. In addition the whole theory of solutions for primal and dual LP, such as the weak and the strong duality theorem, can be generalized for SDP, and this makes the analogy with LP all the more striking. Hence, as in LP, primal and dual SDP have a solution iff their combined KKT conditions

$$\begin{aligned} A^T(y) + S &= C, \\ A(X) &= b, \\ XS &= 0, \\ X \geq 0, S &\geq 0 \end{aligned} \quad (4.17)$$

have a solution.

Many important issues in control theory and structural optimization can be formulated as the problem of minimizing the maximum eigenvalue of an affine matrix which is a SDP problem [60]. Also, relaxations of some important combinatorial problems such as max-cut, bisection, and max clique identification can be formulated as SDPs [24]. Hence, if a polynomial interior-point method for SDP exists, then these NP-hard combinatorial problems will have good polynomial (continuous) approximation algorithms which will have far reaching consequences in developing efficient codes that can handle large-scale problems of this type that often appear in practice. Nesterov and Nemirovskii [51] showed that SDP satisfies the self-concordancy condition and therefore it is possible to devise a polynomial interior-point algorithm for this class of problems. This is the reason for the recent explosion of research activity in this area.

The main step in developing an interior-point algorithm for SDP is the application of one step of the modified Newton method MNM to the function

$$F(X, y, S) = \begin{bmatrix} A^T(y) + S - C \\ A(X) - b \\ XS \end{bmatrix}. \quad (4.18)$$

The domain this function  $\Lambda^m \times R^m \times \Lambda^m$  differs from its range  $\Lambda^m \times R^m \times R^{m \times m}$ , since the product  $XS$  is not symmetric in general. Hence, the  $X$  component of the direction, denoted by  $\Delta X$  is usually not symmetric, even when  $X$  and  $S$  are symmetric. Therefore, it is necessary to reformulate the complementarity condition  $XS = 0$  as an equivalent symmetric condition. Zhang [71] has proposed the condition  $H_p(XS) = 0$  where  $H_p$  is the symmetrization operator defined by

$$H_p(M) = \frac{1}{2} (PMP^{-1} + (PMP^{-1})^T), \quad (4.19)$$

where  $P$  is any nonsingular matrix. The direction  $(\Delta X, \Delta y, \Delta S)$  is then obtained by solving

$$\begin{bmatrix} A^T(\Delta y) + \Delta S \\ A(\Delta X) \\ H_p(S\Delta X + X\Delta S) \end{bmatrix} = \begin{bmatrix} R_c \\ r_b \\ -H_p(XS) + \gamma\mu I \end{bmatrix}, \quad (4.20)$$

where  $R_c = C - A^T(y) - \Delta Z$  and  $r_b = b - A(X)$ . It is not completely straightforward to transform equation (4.19) into a system that can be solved by standard matrix factorizations. The transformations depend strongly on the choice of the matrix  $P$  in equation (4.18). Zhang [71] uses  $P = S^{1/2}$  whereas Alizadeh et al. [5] consider  $P = I$ .

The convergence analysis of the interior-point algorithms for SDP is much more difficult than in the case of LP because of the complicated duality structure of the SDP. Nevertheless, polynomial  $O(\sqrt{n} \log(1/\epsilon))$  global convergence has been proven [5], [66] as well as superlinear local convergence [52].



## 5. IMPLEMENTATIONS

In this section some implementation issues of interior-point algorithms for LP will be discussed. A successful code has not only a good underlying algorithm with good theoretical convergence properties but also a good implementation that enhances the algorithm and makes it efficient and easy to use in practice. A thorough and detailed analysis of the implementation issues for interior-point methods for LP can be found in [4] and the references therein.

### Calculation of the search direction

Computationally the most expensive step in algorithm IPM in Section 3 is the calculation of the search direction by solving the system (3.25) that is shown once again here.

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = \begin{bmatrix} r_p \\ r_D \\ r_{XS} \end{bmatrix}, \quad (5.1)$$

where  $r_{XS} = XS - \gamma\mu e$ . The iteration index  $k$  has been omitted for the sake of simplicity. It seems that this system is much larger than the original system in LP. However, note that system (5.1) is in unreduced form and that it can be greatly reduced. Eliminating  $d_s$  and using the notation  $D = S^{-1/2} X^{1/2}$ , we obtain the following form of the system

$$\begin{bmatrix} -D^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} r_D - X^{-1}r_{XS} \\ r_p \end{bmatrix}, \quad (5.2a)$$

$$d_s = X^{-1}(r_{XS} - Sd_x). \quad (5.2b)$$

The above form is known as an augmented form. The reduction can go even further by eliminating  $d_x$  from the system (5.2a)

$$AD^2 A^T d_y = r_p - A(S^{-1} X r_D + S^{-1} r_{XS}), \quad (5.3a)$$

$$d_s = r_D - A^T d_y, \quad (5.3b)$$

$$d_x = S^{-1}(r_{XS} + X d_s). \quad (5.3c)$$

This form is known as the normal equations form.

The normal equations form is used by most codes because the matrix  $AD^2 A^T$  in equation (5.3a) is usually sparse, symmetric, and positive semidefinite. Hence, it can be factored by sparse Cholesky techniques for which algorithms and codes are well developed and available. The success of the implementation of the Cholesky factorization depends on the quality of its analysis phase which is reordering for sparsity. Its goal is to find a permutation matrix  $P$  such that the Cholesky factor of  $PAD^2 A^T P^T$  is the sparsest possible. In practice, heuristics are used to solve this problem since finding an optimal permutation is a NP-hard problem. Two such heuristics, the minimum degree ordering and the minimum local fill-in ordering are

particularly useful for implementations of interior-point methods. However, two major difficulties remain. The first difficulty is that the matrix  $PAD^2A^TP^T$  may be ill-conditioned. This often occurs when free variables are present in the primal LP. The second difficulty is the handling of dense columns in  $A$ . If a column in  $A$  has  $p$  nonzero components, then a  $p \times p$  dense block will appear in  $PAD^2A^TP^T$ . There are several procedures available to overcome these difficulties, however they have limited success.

When the difficulties mentioned above actually occur during the solving of normal equations (5.3), researchers have recently suggested the use of augmented systems (5.2). The matrix that appears in system (5.2a) is symmetric and indefinite (usually sparse). Algorithms and codes for factoring this type of matrix are not as highly developed or as widely available as sparse Cholesky codes. The best known algorithms are the Bunch-Parlett and the Bunch-Kaufman methods. Recently, researchers have been very busy trying to improve these algorithms or suggest new or even better ones. A common characteristic of these algorithms is that they are less sensitive to ill-conditioning and dense columns than the approach using the normal equations form.

Computational experiments have shown the advantage of the augmented system approach over the normal equations approach when dense columns are present, even though a larger system (dimension  $m+n$ ) has to be solved as compared to the normal equations approach (dimension  $n$ ). In the case of very sparse matrices the approach with the normal equations form was superior. A sophisticated combination of both these approaches may be expected in future codes.

### Termination

Unlike the simplex method, interior-point algorithms never find the exact solution of LP but an approximate one. Most codes simply report an approximate solution that satisfies the termination criteria similar to the one described in algorithm IPM in Section 3. However, instead of taking absolute norms of the residuals, they usually calculate the relative ones.

$$\frac{\|r_p\|}{1+\|b\|} \leq \varepsilon, \quad \frac{\|r_D\|}{1+\|c\|} \leq \varepsilon, \quad \frac{|c^T x - b^T y|}{1+|c^T x|} \leq \varepsilon, \quad (5.4)$$

where  $\varepsilon$  is usually  $10^{-8}$ .

As mentioned in Section 2, it is possible to recover the exact optimal basic solution from the approximate solution. These procedures include finite termination procedures and cross-over procedures ([65], [46]). They are not widely implemented in existing codes since their complexity is similar to the complexity of the simplex method.

### Starting points

The algorithm IPM from Section 3 is an infeasible interior-point algorithm that can start from any point  $(x^0, y^0, s^0)$  with  $(x^0, s^0) > 0$  and still converge. However, computational analysis strongly suggests that the initial point should also satisfy two



additional conditions. First, the point should be well centered, that is, the products  $x_i^0, s_i^0$  should be similar for all  $i$ . Second, the point should not be too infeasible, that is, the ratio  $\|(r_p^0, r_D^0)\|/\mu_0$  of infeasibility to the duality measure should not be too large.

A popular heuristic for finding an initial point starts by calculating a point  $(\tilde{x}, \tilde{y}, \tilde{s})$  that is the solution of the two least square problems

$$\min \|x\|^2 \quad \text{s.t. } Ax = b, \quad (5.5)$$

$$\min \|s\|^2 \quad \text{s.t. } A^T y + s = c. \quad (5.6)$$

Namely, vectors  $\tilde{x}, \tilde{s}$  are the vectors of the least norm for which the residuals  $r_p, r_D$  are zero. The starting point is then defined as

$$(x^0, y^0, s^0) = (\tilde{x} + \delta_x e, \tilde{y}, \tilde{s} + \delta_s e), \quad (5.7)$$

where the scalars  $\delta_x, \delta_s$  are calculated using certain formulas that satisfy the above described conditions. For further details see [44].

Prior information is often available about the solution of LP, in the form of a solution of the slightly perturbed problem, or an estimate of the optimal basis. The interior-point method can use this information to construct a "hot" starting point which often leads to convergence in fewer iterations than the "cold" starting points described above. To construct a hot starting point from the estimate of the solution, we add small positive values to the components of  $(x, s)$  that are at or near their lower bound of zero, and possibly make slight adjustments to the large components of  $(x, s)$  as well. These adjustments should be used in order to ensure that the centrality and infeasibility conditions mentioned above are satisfied. Unfortunately the use of "hot" starting points saves only a few iterations of the interior-point methods.

The consequence of the above discussion is that the simplex method remains much more suitable for sensitivity analysis. Hence, when it is necessary to solve a sequence of similar LPs we can use a mixed method. First, we use an interior-point method with optimal basis recovery to solve the first problem, then we switch to the hot-started simplex method to solve the remaining problems.

### Presolving

The formulation of many practical LP problems contain data that is not necessary such as those variables that are defined but never used, zero columns or rows, duplicate rows or columns, a row with a single nonzero element, etc. Most codes (simplex and interior-point codes) include a procedure that is called a presolver whose purpose is to detect and eliminate many of these unnecessary elements from the data, prior to activating the solver, that is, the code that solves the problem. Presolvers are usually implemented by scanning through the rows and columns of  $A$  repeatedly until no more reductions are found. Presolving is generally much less expensive than a single iteration of IPM, so it is almost always beneficial. Detailed information about presolving can be found in [1], [4].

## **Interior-point codes**

Several efficient interior-point codes for the LP have been developed in recent years. Almost all codes are based on Mehrota's predictor-corrector primal-dual algorithm, similar to the PC-IPM algorithm described in Section 3, although they may differ in many of the implementation details. Also, all codes accept input data in the MPS format, which is a standard format developed decades ago to allow the transfer of input data between different codes based on the simplex method.

Some codes are callable; that is, the user can call them as a subroutine after having filled in the appropriate data structures. The main appeal of callable codes is that they are easily incorporated into larger software packages. Some codes have taken this a step farther; they are available via modeling language interfaces. Modeling languages such as AMPL and GAMS allow users to define their models and data in intuitive terms, defining data structures, variable names, and so on, in a way that naturally fits their application. The modeling language then does the hard work of converting the user-defined model into a format acceptable to the LP code. After a solution is found, it inverts the process, expressing the output from this code in terms of the user's original model. More information on AMPL can be found at (<http://achille.research.att.com/ampl/>) and on GAMS at (<http://www.gams.com/>). The latest development is that some codes allow users access to them over the Internet. Hence, the downloading of the codes is not necessary any more. One such example is Wright's PCx code developed at Argonne National Laboratories. It is available, along with various other codes, at <http://www.mcs.anl.gov/home/otc/Server/> and PCx is in the directory lp.

There are several commercial codes available as well as the numerous public domain research codes. Below, we have briefly described some of the codes in each group.

### **CPLEX(CPLEX/BARRIER)**

**Language:** C

**Algorithm:** Modified Mehrota's predictor-corrector with higher-order corrections, if requested

**Presolving:** Yes

**Input:** MPS, CPLEX format, AMPL, GAMS, binary file, callable

**Vertex solution:** Yes, if requested

**Linear algebra:** Sparse Cholesky factorization of normal equations with procedures to handle ill-conditioning and dense columns

**Availability:** Commercial (CPLEX)

**Additional information:** <http://www.cplex.com/barsolv.html>

### **OSL/EKKBSLV**

**Language:** Fortran

**Algorithm:** Primal barrier, Mehrota's predictor-corrector

**Presolving:** Offered in a separate OSL routine, EKKPRSL

**Input:** MPS, GAMS, callable, a spreadsheet

**Vertex solution:** Yes, if requested



**Linear algebra:** Sparse Cholesky factorization of normal equations with procedures to handle ill-conditioning and dense columns

**Availability:** Commercial (IBM). Free benchmarking (subject to size restriction) is available through the Internet at <http://www.research.ibm.com/osl/bench.html/>

**Additional information:** <http://www.research.ibm.com/osl/>

#### **XPRESS-MP**

**Language:** C and Fortran

**Algorithm:** Mehrota's predictor-corrector on the homogeneous self-dual model with higher-order corrections, if requested

**Presolving:** Yes

**Input:** MPS, GAMS, the binary file from the XPRESS-MP modeler

**Vertex solution:** Yes, if requested

**Linear algebra:** Sparse Cholesky factorization of the normal equations with procedures to handle ill-conditioning and dense columns

**Availability:** Commercial (DASH)

**Additional information:** <http://www.dash.co.uk>

#### **BPMPD**

**Language:** Fortran

**Algorithm:** Modified Mehrota's predictor-corrector with higher-order corrections, if requested

**Presolving:** Yes

**Input:** MPS

**Vertex solution:** No

**Linear algebra:** Sparse Cholesky factorization of the normal equations with procedures to handle ill-conditioning and dense columns

**Availability:** Public domain

**Additional information:** Created by Meszaros,

<ftp://ftp.sztaki.hu/pub/oplab/SOFTWARE/BPMPD>

#### **LIPSOL**

**Language:** Fortran (for sparse Cholesky), MATLAB (for the rest)

**Algorithm:** Modified Mehrota's predictor-corrector

**Presolving:** Some

**Input:** MPS, MATLAB binary, LPP (LP-Plain, a simple format suitable for small problems)

**Vertex solution:** No

**Linear algebra:** Sparse Cholesky factorization of the normal equations with procedures to handle ill-conditioning and dense columns

**Availability:** Public domain

**Additional information:** Created by Zhang,

<http://www.math.umbc.edu/~zhang/lipsol>

## HOPDM

**Language:** Fortran

**Algorithm:** Modified Mehrota's predictor-corrector with higher-order corrections, if requested

**Presolving:** Yes

**Input:** MPS, callable

**Vertex solution:** No

**Linear algebra:** Sparse Cholesky factorization of the normal equations with procedures to handle ill-conditioning and dense columns

**Availability:** Public domain

**Additional information:** Created by Gondzio,

<http://ecolu-info.unige.ch/~logilab/software/hopmd.html>

## LOQO

**Language:** C

**Algorithm:** Mehrota's predictor-corrector

**Presolving:** No

**Input:** MPS, AMPL, GAMS, callable

**Vertex solution:** No

**Linear algebra:** Augmented system factorization with procedures to handle ill-conditioning and dense columns

**Availability:** Free for research purposes, a fee for commercial use

**Additional information:** Created by Vanderbei. Also solves convex QP and general convex programming problems. <http://www.princeton.edu/~rvdb/>

## PCx

**Language:** Fortran (for sparse Cholesky), C (for the rest)

**Algorithm:** Mehrota's predictor-corrector

**Presolving:** Yes

**Input:** MPS, callable, Internet access

**Vertex solution:** No

**Linear algebra:** Sparse Cholesky factorization of the normal equations with procedures to handle ill-conditioning and dense columns

**Availability:** Public domain.

**Additional information:** Created by Czyzyk, Mehrota, and Wright

<http://www.mcs.anl.gov/home/otc/Library/PCx/>

The above codes have been tested on a standard set of LP problems called Netlib (use anonymous ftp to netlib.att.com (cd netlib/lp)). Some of them have also been tested on another set of larger and more difficult LP problems, available at the University of Iowa (via anonymous ftp to col.biz.uiowa.edu (cd pub/testprob/lp)). These and other extensive numerical experiments, along with comparisons with the best LP codes based on the simplex algorithm have shown that good interior-point codes are much more efficient than the best simplex codes on many if not all of the LP problems [4].



## 6. CONCLUSION

The rebirth of interior-point methods that followed the appearance of Karmarkar's paper has had significant theoretical and practical consequences.

The first theoretical consequence is the unified treatment of linear and nonlinear problems via the use of the modified Newton method. In the past, LP problems were analyzed using the simplex method (which is also combinatorial in nature) while nonlinear problems were analyzed using continuous methods that were very often based on Newton-like methods. The second theoretical consequence is a very good global and local convergence of interior-point methods for LP. These results were then successfully extended to many classes of nonlinear programming problems.

One practical consequence is the development of efficient interior-point codes that usually work much better than the best simplex codes. There are two key elements that have contributed to their success. The first element is a good underlying algorithm. The second element is the heavy use of modern matrix factorization algorithms that have been modified to handle ill-conditioning and density. This is an excellent example of how advancements in one field (matrix factorizations) enabled success in another field (interior-point methods) and then further development in the latter field initiates new research in the former field.

Interior-point methods are a very active research area and will remain so for years to come. This research concentrates on the development of efficient interior-point algorithms for many classes of NP problems. The accompanying codes are still not as developed as those for LP and their development is an ongoing process. Recently, there has been a lot of research in the field of semidefinite programming problems because of their applicability to many important combinatorial problems.

A wealth of material on interior-point methods is available in [51], [66], [62], [59], to name but a few. Up-to-date information on interior-point research is available from Interior Point Online, a www site maintained at Argonne National Laboratories at <http://www.mcs.anl.gov/home/otc/InteriorPoint/>. General information about LP, including software information for simplex-base codes and codes for network and integer programming can be found at the same site <http://www.mcs.anl.gov/home/otc/Guide/faq/>.

## REFERENCES

- [1] E. D. Andersen and K. D. Andersen. Presolving in mathematical programming. *Mathematical Programming*, 71, 1995, pp. 221-245.
- [2] M. Anitescu, J. F. Cremer and F. Potra. Formulating 3D contact dynamics problems. *Reports on Computational Mathematics 80*, Department of Mathematics, University of Iowa, Iowa City, 1995.
- [3] I. Adler and N. Megiddo. A simplex algorithm whose average number of steps is bounded between two quadratic functions of the smaller dimension. *Journal of the Association of Computing Machinery*, 32, 1985, pp. 891-895.

- [4] E. D. Andersen, J. Gondzio, C. Mészáros and X. Xu. Implementation of interior-point methods for large scale linear programming. In T. Terlaky, ed. *Interior Point Methods in Mathematical Programming*. Kluwer, 1996.
- [5] F. Alizadeh, J. P. A. Haberly and M. L. Overton. Primal-dual interior-point methods for semidefinite programming; Convergence rates, stability, and numerical results. *Technical Report, Computer Science Department*, Courant Institute of Mathematical Sciences, New York University, New York, 1996.
- [6] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5, 1995, pp. 13-51.
- [7] M. Anitescu, G. Lesaja and F. Potra. An infeasible interior-point predictor-corrector algorithm for the P\*-geometric LCP. *Applied Mathematics and Optimization*, 36, 1997, pp. 203-228.
- [8] M. Anitescu, G. Lesaja and F. Potra. Equivalence between different formulations of the linear complementarity problem. *Optimization Methods and Software*, 7, 3-4, 1997, pp. 265-290.
- [9] K. M. Anstreicher. On long step path following and SUMT for linear programming. *SIAM Journal on Optimization*, 6, 1996, pp. 33-46.
- [10] E. D. Andersen and Y. Ye. On a homogeneous algorithm for the monotone complementarity problem. *Technical Report*, Department of Management Sciences, University of Iowa, Iowa City, 1995.
- [11] R. Bland, D. Goldfarb and M. Todd. The ellipsoid method: a survey. *Operations Research*, 29, 6, 1981, pp. 1039-1091.
- [12] E. R. Barnes. A variation on Karmarkar's algorithm for solving linear programming problems. *Mathematical Programming*, 36, 1986, 174-182.
- [13] K. H. Borgward. *The Simplex Method, A Probabilistic Analysis*. Springer-Verlag, Berlin, 1987.
- [14] R. W. Cottle, J. S. Pang and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, San Diego, 1992.
- [15] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Mathematical Doklady*, 8, 1967, pp. 674-675.
- [16] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, N.J., 1963.
- [17] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York, 1968.
- [18] K. R. Frisch. The logarithmic potential method of convex programming. *Technical Report*, University Institute of Economics, Oslo, 1955.
- [19] M. Grotshel, L. Lovász and A. Shrijver. *The Ellipsoid Method and Combinatorial Optimization*. Springer-Verlag, Bonn, 1986.
- [20] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin and M. H. Wright. On the projected Newton barrier methods for linear programming and an equivalence to



- Karmarkar's projective method. *Mathematical Programming*, 36, 1986, pp. 183-209.
- [21] A. J. Goldman and A. W. Tucker. Theory of linear programming. In H. W. Kuhn and A. W. Tucker, eds. *Linear Equalities and Related Systems*. Princeton University Press, Princeton, N. J., 1956, pp. 53-97.
- [22] O. Guler and Y. Ye. Convergence behavior of interior-point algorithms. *Mathematical Programming*, 60, 1993, pp. 215-228.
- [23] P. Huard. Resolution of mathematical programming with nonlinear constraints by method of centers. In J. Abadie, ed. *Nonlinear Programming*. North Holland, Amsterdam, 1967.
- [24] C. Helmberg, F. Rendl, R. J. Vandrbei and H. Wolkovicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6, 1996, pp. 342-361.
- [25] P. Hung and Y. Ye. An asymptotical  $O(\sqrt{n}L)$ -iteration path-following linear programming algorithm that uses wide neighborhoods. *SIAM Journal on Optimization*, 6, 3, 1996, pp. 570-586.
- [26] F. Jarre. Interior-point methods for convex programming. *Applied Mathematics and Optimization*, 26, 1992, pp. 287-311.
- [27] B. Jansen, K. Roos, T. Terlaky and A. Yoshise. Polynomiality of primal-dual affine scaling algorithms for nonlinear complementarity problems. *Technical Report 95-83*, Faculty of Technical Mathematics and Computer Science, Delft University, Delft, 1995.
- [28] F. Jarre ad M. Saunders. A practical interior-point method for convex programming. *SIAM Journal on Optimization*, 5, 1995, pp. 149-171.
- [29] L. G. Khachiya. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20, 1979, pp. 191-194.
- [30] V. Klee and G. J. Minty. How good is the simplex algorithm?. In O. Shisha, ed. *Inequalities*. Academic Press, New York, 1972, pp. 159-175.
- [31] N. Karmarkar. A polynomial-time algorithm for linear programming. *Combinatorica*, 4, 1984, pp. 373-395.
- [32] M. Kojima, N. Meggido and S. Mizuno. A primal-dual exterior point algorithm for linear programming. *Research Report RJ 8500*, Almaden Research Center, San Jose, 1995.
- [33] M. Kojima, N. Meggido and S. Mizuno. A general framework of continuation methods for complementarity problems. *Mathematics of Operations Research*, 18, 4, 1993, pp. 945-963.
- [34] M. Kojima, N. Meggido T. Noma, and A. Yoshise. *A Unified Approach to Interior-point Algorithm for Linear Complementarity Problems*. Lecture Notes in Computer Science 538, Springer-Verlag, Berlin, 1991.
- [35] M. Kojima, S. Mizuno and A. Yoshise. A primal-dual interior point algorithm for linear programming. In N. Meggido, ed. *Progress in Mathematical*

- programming: *Interior-point Algorithms and Related Methods*. Springer-Verlag, 1989, pp. 29-47.
- [36] M. Kojima, S. Mizuno, and A. Yoshise. An  $O(\sqrt{n}L)$  iteration potential reduction algorithm for linear complementarity problems. *Mathematical Programming*, 50, 1991, pp. 331-342.
- [37] E. Kranich. Interior point methods for mathematical programming: A bibliography. *Discussionbeitrag 171*, FerUniversitat Hagen, Hagen, 1991, (available through Netlib).
- [38] I. J. Lustig, R. E. Marsten and D. Shanno. Interior-point methods for linear programming: Computational state of the art. *ORSA Journal on Computing*, 6, 1994, pp. 1-14.
- [39] G. Lesaja. *Complexity of Linear Programming Problem*. M. S. thesis, University of Zagreb, Zagreb, 1987.
- [40] G. Lesaja. *Interior-Point Methods for  $P^*$ -Complementarity Problems*. Ph.D. thesis, University of Iowa, Iowa City, 1996.
- [41] F. A. Lootsma. Hessian matrices of penalty functions for solving constrained optimization problems. *Philips Research Reports*, 24, 1969, pp. 322-331.
- [42] S. Mehrota. Higher order methods and their performance. *Technical Report 90-16R1*, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, 1991.
- [43] S. Mehrota. Quadratic convergence in primal-dual methods. *Mathematics of Operations Research*, 18, 3, 1993, pp. 741-751.
- [44] S. Mehrota. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2, 1992, pp. 575-601.
- [45] N. Meggido. Pathways to the optimal set in linear programming. In N. Meggido, ed., *Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods*. Springer-Verlag, 1989, pp. 131-158.
- [46] N. Meggido. On finding primal and dual optimal bases. *ORSA Journal on Computing*, 3, 1991, pp. 63-65.
- [47] O. L. Mangasarian. *Nonlinear Programming*. McGraw-Hill, New York, 1969.
- [48] K. G. Murty. Computational complexity of complementary pivot methods. *Mathematical Programming Study*, 7, 1978, pp. 61-73.
- [49] S. Mizuno, M. Todd and Y. Ye. On adaptive step primal-dual interior-point algorithms for linear programming. *Mathematics of Operations Research*, 18, 1993, pp. 964-981.
- [50] W. Murray. Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions. *Journal of Optimization Theory and Applications*, 7, 1971, pp. 189-196.
- [51] Y. E. Nesterev and A. S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming*. SIAM Publications, Philadelphia, 1994.
- [52] F. Potra and R. Sheng. A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming. *Report on Computational*



- Mathematics* 78, Department of Mathematics, University of Iowa, Iowa City, 1995.
- [53] F. Potra and Y. Ye. Interior-point methods for nonlinear complementarity problems. *Reports on Computational Mathematics* 15, Department of Mathematics, University of Iowa, Iowa City, 1991.
- [54] J. Renegar. A polynomial-time algorithm, based on Newton's method, for linear programming. *Mathematical Programming*, 40, 1988, pp. 59-73.
- [55] N. Z. Shor. Cut-off method with space extensions in convex programming. *Kibernetika*, 6, 1987, pp. 94-95.
- [56] S. Smale. On the average number of steps of the simplex method of linear programming. *Mathematical Programming*, 27, 1983, pp. 241-262.
- [57] G. Sonnevend. An analytical center for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In *Lecture Notes in Control and Information Sciences* 84, Springer-Verlag, Berlin, 1986, pp. 866-876.
- [58] J. Sun and G. Zhao. A quadratically convergent polynomial long-step algorithm for the class of nonlinear complementarity problems. *Working Paper*, National University of Singapore, 1995.
- [59] T. Terlaky, ed. *Interior Point Methods of Mathematical Programming*. Kluwer, 1996
- [60] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38, 1996, pp. 49-95.
- [61] R. J. Vanderbei, M. S. Meketon and B. A. Friedman. A modification of Karmarkar's linear programming algorithm. *Algorithmica*, 1, 4, 1986, pp. 395-407.
- [62] S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM Publications, Philadelphia, 1997.
- [63] X. Xu, P. Hung and Y. Ye. A simplified homogeneous and self-dual linear programming algorithm and its implementation. *Annals of Operations Research*, 62, 1996, pp. 151-172.
- [64] Y. Ye. An  $O(n^3L)$  potential reduction algorithm for linear programming. *Mathematical Programming*, 50, 1991, pp. 239-258.
- [65] Y. Ye. On the finite convergence of interior-point algorithms for linear programming. *Mathematical Programming*, 57, 1992, pp. 325-336.
- [66] Y. Ye. *Interior-Point Algorithm: Theory and Analysis*. John Wiley and Sons, 1997.
- [67] Y. Ye and K. Anstracher. On quadratic and  $O(\sqrt{n}L)$  convergence of a predictor-corrector algorithm for LCP. *Mathematical Programming*, 62, 1993, pp. 537-551.

- [68] Y. Ye, O. Guler, R. A. Tapia and Y. Zhang. A quadraticall convergent  $O(\sqrt{nL})$ -iteration algorithm for linear programming. *Mathematical Programming*, 59, 2, 1993, pp. 151-162.
- [69] H. Yamashita. A polynomially and quadratically convergent method for linear programming. *Research Report*, Mathematical Systems Institute, Tokyo, 1986.
- [70] Y. Ye, M. J. Todd and S. Mizuno. An  $O(\sqrt{nL})$ -iteration homogeneous and self-dual linear programming algorithm. *Mathematics Operations Research*, 19, 1994, pp. 208-227.
- [71] Y. Zhang. On extending primal-dual interior-point algorithm from linear programming to semidefinite programming. *Technical Report TR95-20*, Department of Mathematics and Statistics, University of Maryland, Baltimore, 1995.
- [72] Y. Zhang and R. A. Tapia. A quadratically convergent polynomial primal-dual interior point algorithm for linear programming. *SIAM Journal on Optimization*, 1, 1993.

Received: 22 October 1999

Accepted: 25 November 1999

Goran Lešaja

## METODE UNUTARNJE TOČKE I SUVREMENA PROGRAMSKA PODRŠKA ZA OPTIMALIZACIJU

### Sažetak

U radu su prikazane glavne metode za optimalizaciju koje se temelje na metodi unutarnje točke. Prikazani su ključni elementi metoda unutarnje točke za problem linearnog programiranja, a dana su i njihova proširenja za probleme nelinearnog programiranja. Opisani su i najpoznatiji suvremeni programski paketi za optimalizaciju u kojima se koriste metode unutarnje točke i naglašava se da su za probleme linearnog programiranja velikih dimenzija ti programi često efikasniji od onih koji se temelje na simpleks metodi.

**Ključne riječi:** metode unutarnje točke, optimizacijski softveri, Newtonova metoda, linearno programiranje, nelinearno programiranje, konveksno programiranje, semidefinitno programiranje, algoritam Mehrota.