3

A computational view of interior point methods

Jacek Gondzio

Logilab, HEC Geneva,
Section of Management Studies, University of Geneva,
102 Bd Carl Vogt, CH-1211 Geneva 4, Switzerland

Tamás Terlaky

Faculty of Technical Mathematics and Informatics,
Delft University of Technology,
P.O. Box 5031, 2600 GA Delft, The Netherlands

1 Overview

Many issues that are crucial for an efficient implementation of an interior point algorithm are addressed in this chapter. To start with, a prototype primal-dual algorithm is presented. Next, many tricks that make it efficient in practice are discussed in detail. Those include: preprocessing techniques, initialization approaches, methods for computing search directions (and the underlying linear algebra techniques), centering strategies and methods of stepsize selection.

Several reasons for the manifestation of numerical difficulties, for example the primal degeneracy of optimal solutions or the lack of feasible solutions, are explained in a comprehensive way.

A motivation for obtaining an optimal basis is given and a practicable algorithm to perform this task is presented. Advantages of different methods to perform postoptimal analysis (applicable to interior point optimal solutions) are discussed.

Important questions that still remain open in the implementation of interior point methods are also addressed, e.g. performing correct postoptimal analysis, detecting infeasibility or resolving difficulties arising in the presence of unbounded optimal faces. The challenging practical problem of warm start is recalled and two potentially attractive approaches to it are suggested.

To facilitate understanding of different implementation strategies, some illustrative numerical results on a subset of problems from the NETLIB collection are presented.

The research of Jacek Gondzio has been partially supported by the Committee for Scientific Research of Poland, grant No PB 8 S505 015 05 and by the Fonds National de la Recherche Scientifique Suisse, grant No 12-34002.92.
The research of Tamás Terlaky has been partially supported by OTKA No 2116.

2 Introduction

Karmarkar's publication in 1984 of a new polynomial time algorithm for linear programming (LP) [40] drew enormous attention from the mathematical programming community and has led to great activity, resulting in a flood of papers (see, e.g. [45]).

The idea of crossing the interior of the feasible region in search of an optimum of the linear program was present at least since the 1960s. These methods were for example: an affine-scaling method of Dikin [18] and a logarithmic barrier method SUMT of Fiacco and McCormick [22]. For at least two reasons, however, these methods could not at the time be shown to be competitive to simplex. First, due to the storage limitations, the size of the problems solved in the late 1960s never exceeded several hundred rows and columns and for such sizes the simplex method is practically unbeatable. Secondly, there were no sparse symmetric solvers available at that time (they appeared at the beginning of the 1970s) so the orthogonal projections must have killed the efficiency of interior point methods (IPMs). IPMs need significantly more memory than the simplex method which was an unacceptable requirement at that time.

Clearly, the situation was quite different in 1984, which encouraged Karmarkar to claim an excellent efficiency for his new approach. In fact, these claims still had to wait several years to be confirmed by computational results [1, 2, 59].

Soon after Karmarkar's publication, Gill, Murray, Saunders, Tomlin and Wright [27] built the bridge between this new interior point method and the logarithmic barrier approach. Barrier methods were developed for the primal and for the dual LP formulation (see, e.g. the surveys [32, 65, 67]). Early implementations that were based on pure primal or dual methods gave competitive results with simplex implementations. Nowadays all the state-of-the-art IPM implementations are those of primal-dual methods, hence in this chapter we concentrate only on primal-dual methods.

First Megiddo [52] proposed applying a logarithmic barrier method to the primal and the dual problems at the same time. Independently, Kojima, Mizuno and Yoshise [44] developed the theoretical background of this method and gave complexity results. Its early implementations [15, 51] showed much promise and encouraged further research. For extensions that represent current state-of-the-art primal-dual implementations, see Lustig, Marsten and Shanno [49, 47, 48] and Mehrotra [54, 55].

A primal-dual algorithm is a feasible IPM if all the iterates are primal and dual feasible. If the iterates are positive but infeasible then the primal-dual algorithm is called an *infeasible IPM*. This algorithm attains feasibility at the same time as optimality is reached. It had been successfully implemented that way [47] and had shown very good practical convergence long before a theoretical justification for such behaviour was found by Kojima, Megiddo and Mizuno [43]. The method has proven polynomial complexity: $O(n^2L)$ in [79] and $O(nL)$ in [57, 62].

Although the complexity of the infeasible primal-dual algorithm is worse than the best-known complexity $O(\sqrt{n}L)$ of most feasible IPMs (see, e.g. the surveys [32, 65, 67]), it is now widely accepted that primal-dual infeasible IPMs are more efficient in implementations. Since infeasible IPMs are the methods of choice to date for "state of the art" implementations, throughout the whole chapter we mean a *primal-dual infeasible IPM* we speak about a primal-dual algorithm. To facilitate this, in Section 3 we shall introduce a prototype primal-dual infeasible IPM algorithm.

A common feature of almost all IPMs is that they can be interpreted in terms of following the path of centers [68] that leads to the optimal solution (see, e.g. [32, 65]). With some abuse of mathematics, a basic iteration of a path-following algorithm consists of moving from one point in a neighbourhood of the central path to another one called the *target* that preserves the property of lying in a neighbourhood of the central path and reduces the distance to optimality measured with some estimation of the duality gap. Such a movement can in principle involve more than one step towards the target. Depending on how significant is the update of the target (and consequently, whether just one or more Newton steps are needed to reach the vicinity of the new target) one distinguishes between short-step and long-step methods. Due to the considerable cost of every Newton step, usually (at least in implementations) one Newton step is allowed before a new target is defined.

Every Newton step requires computing at least one orthogonal projection onto the null space of a scaled linear operator AD, where A is the LP constraint matrix and D is a positive diagonal scaling matrix that changes in subsequent iterations. Primal, dual and primal-dual variants of IPMs differ on the way matrix D is defined, but the effort to compute Karmarkar's projection is always the same. Every orthogonal projection involves inversion of the matrix AD^2A^T – the most time consuming linear algebra operation which takes about 60–90% of the computation time of a single interior point iteration. Unless the linear program is specially structured and this structure can be exploited to determine an easily invertible preconditioner for an iterative method (such as e.g. a conjugate gradient algorithm, implemented successfully for network problems [61, 64]), direct methods [19] that compute a sparse symmetric factorization (Cholesky decomposition of the positive definite system AD^2A^T or Bunch–Parlett [7, 13] decomposition of the indefinite augmented system $\begin{bmatrix} D^{-2} & A^T \\ A & 0 \end{bmatrix}$) are the methods of choice. Computing projections onto affine spaces seems crucial for the efficiency of any interior point algorithm. We shall thus discuss it in detail in Section 4 which also addresses other issues of implementation of the IPM such as the role of presolve analysis, the choice of the starting point, the choice of the stepsizes in the primal and in the dual spaces, the role of centering, higher-order methods, the termination conditions and, finally, the comparison of theoretical and practical complexity.

In Section 5 we shall add some remarks relating to manifestations of degeneracy and ill-conditioning in the computation of projections.

For about forty years, the simplex method (starting from its discovery in 1947 [16] until Karmarkar's breakthrough [40]) was the only effective algorithm for linear programs. Hence, operations research practitioners got used to seeing linear programming from the simplex perspective. This, in particular, applies to the use of the postoptimality analysis available from the optimal basis solution. In fact, such a postoptimality analysis is almost always incomplete (frequently incorrect), see e.g. [17, 33, 37]. Nevertheless, there exist many applications in which an optimal basis is necessary, e.g. reoptimization in integer programming. In such a case a need arises for identifying an optimal basis from the interior point optimal solution. Fortunately, this can be done in strongly polynomial time [53]. We shall address the problem of optimal basis identification in Section 6.

Section 7 will be devoted to some crucial questions that still remain open. Sensitivity analysis based on interior point optimal solution is generally more expensive but produces correct information. We discuss how to handle problems with unbounded level sets, how to detect infeasibility and how to implement efficient warm start in interior point algorithms.

Most relevant issues of interior point method implementations will be illustrated by solving a subset of the NETLIB LP problem test collection using version 2.0 of the HOPDM (Higher Order Primal Dual Method) code [4, 31]. All of our computations are made on a SUN SPARC-10 workstation.

Later on in the chapter we will frequently speak about stability, robustness and efficiency of different methods. On stability, the usual numerical stability is meant. Talking about robustness, one thinks about whether the algorithm gives reliable answers on a wide range (optimally all) of problem instances. Finally, efficiency relates to the speed of the algorithm, and the speed of the implementation.

3 A prototype primal-dual algorithm

Let us consider a primal linear programming problem:

$$\begin{array}{ll} \text{minimize} & c^T x, \\ \text{subject to} & Ax = b, \\ & x + s = u, \\ & x, s \geq 0, \end{array} \tag{3.1}$$

where $c, x, s, u \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m*n}$; and its dual:

$$\begin{array}{ll} \text{maximize} & b^T y - u^T t, \\ \text{subject to} & A^T y - t + z = c, \\ & z, t \geq 0, \end{array} \tag{3.2}$$

where $y \in \mathbb{R}^m$ and $z, t \in \mathbb{R}^n$. Here we assume that rank $A = m$ (in Section 4.1.3 we will see that this assumption is not restrictive).

To derive the primal-dual algorithm, let us replace non-negativity of con-