Journal of Computational Mathematics
Vol.31, No.4, 2013, 335–354.
http://www.global-sci.org/jcm
doi:10.4208/jcm.1301-m4106

ON AN EFFICIENT IMPLEMENTATION OF THE FACE ALGORITHM FOR LINEAR PROGRAMMING*

Lei-Hong Zhang
Department of Applied Mathematics, Shanghai University of Finance and Economics, Shanghai 200433, China
Email: longzlh@gmail.com
Wei Hong Yang
School of Mathematical Sciences, Fudan University, Shanghai 200433, China
Email: whyang@fudan.edu.cn
Li-Zhi Liao
Department of Mathematics and Institute of Computational and Theoretical Studies, Hong Kong Baptist University, Hong Kong, China
Email: liliao@hkbu.edu.hk

Abstract

In this paper, we consider the solution of the standard linear programming (LP). A remarkable result in LP claims that all optimal solutions form an optimal face of the underlying polyhedron. In practice, many real-world problems have infinitely many optimal solutions and pursuing the optimal face, not just an optimal vertex, is quite desirable. The face algorithm proposed by Pan [19] targets at the optimal face by iterating from face to face, along an orthogonal projection of the negative objective gradient onto a relevant null space. The algorithm exhibits a favorable numerical performance by comparing the simplex method. In this paper, we further investigate the face algorithm by proposing an improved implementation. In exact arithmetic computation, the new algorithm generates the same sequence as Pan's face algorithm, but uses less computational costs per iteration, and enjoys favorable properties for sparse problems.

Mathematics subject classification: 62H20, 15A12, 65F10, 65K05.
Key words: Linear programming, Level face, Optimal face, Rank-one correction.

1. Introduction

In this paper, we consider the solution of the linear programming (LP) in the standard form:

$$\begin{array}{ll} \min & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \end{array} \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}$ $(m < n)$, and $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$.

There are two basic classes of algorithms for solving LP in the literature. The first milestone is the well-known simplex method founded by Dantzig [2], in which the philosophy is to move on the underlying polyhedron, from vertex to an adjacent vertex, until reaching an optimal vertex. Since then, extensive theoretical analysis, numerical implementations as well as numerous variants (see e.g., [1,3,4,6,7,9,13–18,20–24]) have been developed. To date, the simplex algorithm is accepted as one of the most famous and widely used mathematical tools in the world [13]. The

* Received March 22, 2012 / Revised version received December 29, 2012 / Accepted January 31, 2013 / Published online July 9, 2013 /

other type of methods distinguishes the simplex method by approaching an optimal solution (not necessarily an optimal vertex) from inside of the polyhedron, and they are usually categorized as the interior point methods. Karmarkar's projective algorithm [10] is one of successful and practical interior point methods and stimulates this trend (see e.g., [11–13, 23]). Both the simplex method (and its variants) and the interior point methods have their own advantages and disadvantages; in particular, some interior point methods can reach an optimal solution in polynomial time theoretically, yet neither of them in general shows a dominant performance to the other numerically.

One of the remarkable results for LP (1.1) is that whenever (1.1) has a solution, then at least one vertex of the polyhedron of (1.1) is an optimal solution (see e.g., [13]). This serves as the theoretical fundamental for Dantzig's simplex method. Another well-known result of (1.1) claims that all the optimal solutions form an optimal *face* (see Definition 2.1). In practice, many real-world problems usually have infinitely many optimal solutions and pursuing an optimal face, not just a vertex (a vertex is a special face), is quite desirable. This motives the face algorithm proposed by Pan [19, Chapter 8][1)] . The basic idea behind the face algorithm is to move from face to face, along an orthogonal projection of the negative objective gradient onto a relevant null space (the search direction), to reach an optimal face. The search direction can be efficiently computed via the Cholesky factorization. Preliminary computational testing is carried out and the results show that the face algorithm has a favorable numerical performance [19, Chapter 8].

In this paper, we further investigate the face algorithm by proposing an improved implementation. The basic idea of our new implementation is to *update* the search direction in a similar manner as the *eta-matrix* technique proposed by Dantzig and Orchard-Hayes [3] for the simplex method. The key for our implementation is that the kth search direction of the face algorithm is the solution of a relevant linear system whose coefficient matrix only has a rank-one correction upon that of $(k-1)$th. This fact, with the aid of the Sherman-Morrison formula (see Lemma 4.1), then makes it possible to reuse the previous information and then update the search direction in a similar fashion to the *eta-matrix* technique of Dantzig and Orchard-Hayes to improve the efficiency of the face algorithm. In exact arithmetic computation, we will see that our improved face algorithm generates the same sequence as the face algorithm of Pan [19, Chapter 8], but the new implementation uses less computational costs per iteration and enjoys favorable properties for sparse problems. The detailed computational gains and performance tradeoffs will be discussed in Section 5.

The remaining paper is organized in the following way. In Section 2, some basic concepts and results related to the face algorithm are provided. Section 3 then outlines the original face algorithm proposed in [19, Chapter 8]. Our new implementation, together with some new and further properties of the face algorithm, is presented in Section 4. The computational gains as well as performance tradeoffs of our new algorithm are then discussed in Section 5. Finally, we report computational results in Section 6 and draw a conclusion in Section 7.

Throughout the paper, all vectors are column vectors and are typeset in bold. For a given vector $\mathbf{x}$, we use x_i to denote the i-th component of $\mathbf{x}$. For a matrix $A \in \mathbb{R}^{n\times m}$, $A^\top$ denotes its transpose, and $\mathrm{Null}(A)$ stands for the null space of A.

[1)] The face algorithm is also presented and available at: P.-Q. Pan, *A face algorithm for linear programming,* preprint. http://www.optimization-online.org/DB_HTML/2007/10/1806.html.