European Journal of Operational Research 245 (2015) 371–383

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Continuous Optimization

A linear programming decomposition focusing on the span of the nondegenerate columns

Jérémy Omer[a,b,*], François Soumis[a,b]

[a] École Polytechnique de Montréal, 2900 bd. Edouard-Montpetit, Montréal, QC H3T 1J4, Canada
[b] Group for Research in Decision Analysis, HEC Montral, 3000 ch. de la Cte-Sainte-Catherine, Montréal, QC H3T 2A7, Canada

ARTICLE INFO

Article history:
Received 28 July 2014
Accepted 12 March 2015
Available online 16 March 2015

Keywords:
Linear programming
Degeneracy
Improved primal simplex
Decomposition
Primal algorithms

ABSTRACT

The improved primal simplex (IPS) was recently developed by Elhalaloui et al. to take advantage of degeneracy when solving linear programs with the primal simplex. It implements a dynamic constraint reduction based on the compatible columns, i.e., those that belong to the span of a given subset of basic columns including the nondegenerate ones. The identification of the compatible variables may however be computationally costly and a large number of linear programs are solved to enlarge the subset of basic variables. In this article, we first show how the positive edge criterion of Raymond et al. can be included in IPS for a fast identification of the compatible variables. Our algorithm then proceeds through a series of reduction and augmentation phases until optimality is reached. In a reduction phase, we identify compatible variables and focus on them to make quick progress toward optimality. During an augmentation phase, we compute one greatest normalized improving direction and select a subset of variables that should be considered in the reduced problem. Compared with IPS, the linear program that is solved to find this direction involves the data of the original constraint matrix. This new algorithm is tested over Mittelmann's benchmark for linear programming and on instances arising from industrial applications. The results show that the new algorithm outperforms the primal simplex of CPLEX on most highly degenerate instances in which a sufficient number of nonbasic variables are compatible. In contrast, IPS has difficulties on the eleven largest Mittelmann instances.

1. Introduction

We consider a linear program (LP) in standard form:

$$\begin{cases} \min & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{cases} \tag{P}$$

where $\boldsymbol{x}, \boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. We assume that $\boldsymbol{A}$ is of full rank m with $m \leq n$ and that the feasible domain $\mathcal{F}_P = \{\boldsymbol{x} \geq \boldsymbol{0} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$ is nonempty. A basis is a set of m independent columns of $\boldsymbol{A}$, and the associated variables are said to be basic. Starting from the indices $\mathcal{B}$ of the basic variables and $\mathcal{N}$ of the remaining nonbasic variables, the associated basic solution is obtained by setting

$$\boldsymbol{x}_{\mathcal{B}} = \boldsymbol{A}_{\cdot\mathcal{B}}^{-1}\boldsymbol{b} \quad \text{and} \quad \boldsymbol{x}_{\mathcal{N}} = \boldsymbol{0}, \tag{1}$$

where for any set of indices $\mathcal{J}$, $\boldsymbol{A}_{\cdot\mathcal{J}}$ is the set of columns of $\boldsymbol{A}$ indexed by $\mathcal{J}$, and $\boldsymbol{x}_{\mathcal{J}}$ is the corresponding subvector of variables. More generally, the submatrix of $\boldsymbol{A}$ containing the rows indexed by $\mathcal{I}$ and the columns indexed by $\mathcal{J}$ will be denoted $\boldsymbol{A}_{\mathcal{I}\mathcal{J}}$. The basic solution is feasible if and only if $\boldsymbol{x}_{\mathcal{B}} \geq \boldsymbol{0}$. If $\{j \in \mathcal{B} : x_j = 0\}$ is not empty, the solution is said to be degenerate, and all the variables indexed by this set are degenerate. The remaining nonzero basic variables are the nondegenerate variables.

1.1. Dealing with degeneracy in the primal simplex

Starting from a basic feasible solution, the primal simplex algorithm (see Dantzig, 1955) monotonically improves the objective value by going through a sequence of neighboring feasible bases until optimality is reached. One theoretical limitation of the algorithm is that an iteration may not lead to any progress in the objective value if the solution is degenerate. Geometrically, a degenerate vertex of the n-dimensional feasible polytope of an LP is the intersection of more than n constraints of this LP. In terms of the simplex algorithm, this means that a single vertex can correspond to several bases. The difficulty is that sometimes many iterations move from one basis to another associated with the same vertex. As a consequence, the theoretical convergence of the simplex cannot be guaranteed without a pivoting rule such as those described in Bland (1977) and Charnes (1952).

* Corresponding author. Tel.: +1 5144339243.
E-mail addresses: jeremy.omer@gmail.com, jeremy.omer@gerad.ca (J. Omer), francois.soumis@gerad.ca (F. Soumis).

http://dx.doi.org/10.1016/j.ejor.2015.03.019
0377-2217/

Although cycling is rarely an issue in practice, the risk of stalling is real. Several techniques have been developed to limit the negative effects of degeneracy (Benichou, Gauthier, Hentges, & Ribiere, 1977; Charnes, 1952; Gal, 1993; Gill, Murray, Saunders, & Wright, 1989; Greenberg, 1978), but recently there has been a growing interest in methods that take advantage of degeneracy. These studies all rely on the idea that degeneracy corresponds to a local excess of information, since degenerate basic variables are not needed to characterize a vertex of the polytope. Perold (1980) exploits this to develop a degeneracy structure in the LU decomposition of the basis, which involves fewer calculations when performing degenerate pivots. Pan (1998) took another important step in this direction by generalizing the concept of a basis. He defines a deficient basis to be a set of less than m independent columns of $\boldsymbol{A}$ whose range contains $\boldsymbol{b}$. If the current solution is degenerate, it is sufficient to consider the deficient basis that contains only the positive variables. Degeneracy therefore becomes a potential opportunity to solve smaller linear systems at each iteration. Using deficient bases, Pan develops a simplex-like algorithm (Pan, 2008) and a dual projective algorithm (Pan, 2005) that show promising results in an experimental comparison with MINOS 5 (Murtagh & Saunders, 1983).

Elhallaoui, Metrane, Soumis, and Desaulniers (2010); Elhallaoui, Villeneuve, Soumis, and Desaulniers (2005) also take advantage of degeneracy to speed up the solution of set partitioning problems by aggregating the original constraints into clusters of constraints. The feasibility of the solution is ensured by keeping only the variables that are compatible with the clusters, i.e., the variables that are either present in or absent from every constraint of each cluster. When no improvement can be made by considering the compatible variables, some clusters are broken up or combined to include new improving directions in the aggregated problem. The strength of this dynamic constraint aggregation is the focus on a problem with many fewer constraints than the original one. The improved primal simplex (IPS) (Elhallaoui, Metrane, Desaulniers, & Soumis, 2011) extends this approach to general linear programming. A reduced problem is formed by keeping only the nondegenerate and compatible variables. In this context, a variable is compatible when the corresponding column of $\boldsymbol{A}$ is in the range of the p nondegenerate columns. With the incompatible variables removed, $m-p$ constraints are redundant and thus ignored. Once the optimal solution of the reduced problem has been found, a complementary problem is solved to prove the optimality of the original LP or to identify a sequence of pivots ending with an improvement in the objective value. The authors report that IPS significantly outperforms CPLEX[1] on flight assignment (FA), combined vehicle and crew scheduling (VCS), and uncapacitated facility location (UFL) problems, and Raymond, Soumis, and Orban (2010) describe implementation techniques that improve the performance of the algorithm.

One important limitation of IPS is that compatible variables are identified through costly algebraic operations similar to those performed when computing a simplex tableau. As highlighted by Omer, Rosat, Raymond, and Soumis (2014), these operations are also useful when solving the complementary problem since they allow us to search for an improving direction in a reduced space, as is done in reduced gradient methods (Murtagh & Saunders, 1978). However, their tests on a diversified benchmark show that these operations cause IPS not to perform well on every highly degenerate LP. Raymond, Soumis, Metrane, and Desrosiers (2010) address this issue with a stochastic test requiring as many operations as the computation of a reduced cost to identify all the compatible variables. The authors apply this test to develop a partial pricing algorithm focusing on the compatible variables first. They report good results on the aforementioned VCS and FA instances, but their procedure struggles with two families of instances represented in Mittelmann's benchmark. Based on this test, Towhidi, Desrosiers, and Soumis (2014) implement the positive edge pricing criterion within COIN-OR LP solver[2] (CLP). Their results show significant improvement with regards to the devex pricing criterion (Harris, 1973) for the most degenerate Mittelmann instances, but their comparison focuses on CLP, which is known to be less efficient than most commercial LP solvers. More importantly, the articles by Towhidi et al. (2014) and Raymond, Soumis, Metrane et al. (2010) show how a fast compatibility test can be used to cope with degeneracy, but they do not take advantage of degeneracy, since the size of the linear system solved at each simplex pivot is not reduced.

1.2. Contribution statement

Although the dual simplex and barrier algorithms often solve LPs more efficiently than the primal simplex, the latter has a strong advantage when a good feasible solution is available. As a consequence, the primal simplex is still used for reoptimization after modifications in the objective function, or after adding columns in the master problem in a column-generation procedure. Our work thus focuses on improving the primal simplex by taking advantage of degeneracy.

Our main contribution is a new dynamic reduction algorithm that overcomes the difficulties that IPS encounters on large instances. This algorithm not only yields substantial improvement on many degenerate instances but also provides a fast procedure to test the potential for improvement in advance. To achieve this, we modify IPS to include the fast compatibility test described in Raymond, Soumis, Metrane et al. (2010). One negative effect is that the complementary problem cannot be reduced without doing the algebraic operations that we are trying to avoid. The algorithm thus focuses on a complementary problem involving the original constraints of P, and it involves a new mode of alternation between the reduced and the complementary problems that is more efficient on large LPs. We then show how good basic solutions can be built to warm-start both the reduced and the complementary problems. The practical impact of these modifications is studied on a large benchmark including the VCS, FA, and UFL instances used in Omer et al. (2014) and 45 Mittelmann instances. The purpose is to evaluate our new algorithm by comparing it with IPS and the primal simplex of CPLEX, and to show that it is possible to identify quickly the instances that offer a strong potential for faster solution. The results show the potential of the algorithm for an implementation as an adaptive strategy in a state-of-the-art primal simplex code.

In Section 2 we describe IPS as a necessary background for the rest of the article. The new algorithm based on a fast compatibility test is developed in Section 3. The results of the experimental tests are presented and analyzed in Section 4, and in Section 5 we discuss directions for future research.

2. The improved primal simplex

In this section, we summarize the theoretical foundations and the practical implementation of IPS as a background for the new algorithm developed in Section 3. Although efficient implementations of linear programming algorithms should focus on LPs with bounded variables, we consider an LP in standard form to clarify and shorten the presentation. Omer et al. (2014) show the generalization to an LP with bounded variables, and the implementations tested in Section 4 use this generalization.

Let $\boldsymbol{x} \in \mathcal{F}_P$ be a basic feasible solution of P. The variables' indices can be partitioned into two sets $\mathcal{P} = \{j : x_j > 0\}$ and $\mathcal{L} = \{j : x_j = 0\}$. Since $\boldsymbol{x}$ is a basic solution, the variables indexed by $\mathcal{P}$ are basic, and

[1] CPLEX is freely available for academic and research purposes under the IBM academic initiative: http://www-03.ibm.com/ibm/university/academic .

[2] https://projects.coin-or.org/Clp .