The positive edge pricing rule for the dual simplex

Jérémy Omer, Mehdi Towhidi, François Soumis

Author's Accepted Manuscript

www.elsevier.com/locate/caor

The positive edge pricing rule for the dual simplex

Jérémy Omer, Mehdi Towhidi, François Soumis,

PII: S0305-0548(15)00067-2
DOI: http://dx.doi.org/10.1016/j.cor.2015.03.009
Reference: CAOR3749

To appear in: *Computers & Operations Research*

Cite this article as: Jérémy Omer, Mehdi Towhidi, François Soumis, , The positive edge pricing rule for the dual simplex, *Computers & Operations Research,* http://dx.doi.org/10.1016/j.cor.2015.03.009

The positive edge pricing rule for the dual simplex

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

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

Abstract

In this article, we develop the two-dimensional positive edge criterion for the dual simplex. This work extends a similar pricing rule implemented by Towhidi et al. [24] to reduce the negative effects of degeneracy in the primal simplex. In the dual simplex, degeneracy occurs when nonbasic variables have a zero reduced cost, and it may lead to pivots that do not improve the objective value. We analyze dual degeneracy to characterize a particular set of dual compatible variables such that if any of them is selected to leave the basis the pivot will be nondegenerate. The dual positive edge rule can be used to modify any pivot selection rule so as to prioritize compatible variables. The expected effect is to reduce the number of pivots during the solution of degenerate problems with the dual simplex. For the experiments, we implement the positive edge rule within the dual simplex of the COIN-OR LP solver, and combine it with both the dual Dantzig and the dual steepest edge criteria. We test our implementation on 62 instances from four well-known benchmarks for linear programming. The results show that the dual positive edge rule significantly improves on the classical pricing rules.

Keywords: Linear programming, Dual simplex, Degeneracy, Pricing criterion, Positive edge

1. Introduction

1.1. Degeneracy in the dual simplex

More than sixty years after it was first introduced by Dantzig (see [4]), the simplex remains one of the most popular algorithms for linear programs (LPs). The algorithm iteratively progresses toward the optimal solution by stepping from one feasible vertex to an adjacent vertex with a better objective value. Lemke [16] developed the dual counterpart of the primal simplex in 1954. The difference between this *dual simplex* and the simplex applied to the dual LP is that the latter keeps the primal representation. In particular, all the computations that occur at each pivot refer to a basis of the primal space. The dual simplex found an early application in mixed-integer linear programming: it can be used to efficiently reoptimize the solution after branching or adding cuts. However, according to Bixby [3], it had to wait for the major contributions of Forrest and Goldfarb [8] and Fourer [9] before becoming a practical alternative to the primal simplex for solving LPs. A tutorial on the dual simplex may be found in [1], and recent progress with respect to implementation issues is surveyed in [15].

One of the difficulties that the simplex may encounter is *degeneracy*. A *degenerate pivot* is performed when the basis changes without improvement in the objective value. In the dual simplex, this situation can arise when nonbasic variables have a zero reduced cost. If one of these *dual degenerate variables* is chosen to enter the basis, then it is impossible to improve the objective value without violating a constraint of the dual problem.

Research into degeneracy has mostly focused on the primal simplex, but the techniques can be adapted for the dual method. For instance, a random perturbation of the constraints' right-hand side [2] and a similar shifting of bounds on the variables [11] are implemented in efficient codes to reduce the negative effects of degeneracy in the primal simplex. These perturbations replace the degenerate pivots with pivots that give a small but guaranteed improvement in the objective value. They are also used in the dual simplex to modify the objective function coefficients [3]. One

*Tel: +1 514 340 5121 #6051
Email address: jeremy.omer@gmail.com (Jérémy Omer)

technique specific to the dual approach is the bound-flipping ratio test [9]. The idea is that bounded variables offer the opportunity for larger steps. In some circumstances, the value of the variable selected to enter the basis can be changed from one of its bounds to the other while maintaining the dual feasibility of the basis. Another variable can then be selected to make another step without additional computational effort. In many situations, this ratio test avoids a degenerate pivot by flipping the bounds of degenerate variables. A detailed description of a bound-flipping dual can be found in [18].

The pricing criterion is also critical in the context of degeneracy. The dual steepest edge algorithms described by Forrest and Goldfarb in [8] motivated practical interest in the dual simplex. The classical Dantzig criterion selects the pivot row that corresponds to the largest infeasibility. The steepest edge criterion considers the ratios of the infeasibilities to the norms of the rows of the basis inverse. This criterion generally produces larger steps and reduces the effects of degeneracy [13].

In the primal case, several attempts have been made to identify the variables that will produce a nondegenerate pivot if they are selected during the pricing step. The exact identification of this set of variables is impossible in practice since it involves the computation of the whole simplex tableau. In [12, 14], a subset of the variables that will produce a degenerate pivot is found, and the number of algebraic operations is similar to that for the computation of the reduced cost vector. This rule improves the likelihood of performing a nondegenerate pivot by discarding a set of bad candidates. In contrast, Raymond et al. [23] developed the *positive edge* test to efficiently determine a set of *compatible* variables that will give rise to a nondegenerate pivot. The variables are those that have their corresponding columns in the span of the nondegenerate columns. Towhidi et al. [24] introduced the positive edge pricing rule to prioritize the compatible variables in the primal simplex. Their implementation of the criterion within the COIN-OR LP (CLP) solver [17] produces good results on the most degenerate instances of Mittelmann's benchmark[1].

1.2. Contribution statement

The most recent research on degeneracy in the simplex has mostly focused on the primal algorithm. In [22], Pan describes a dual projective simplex for degenerate problems. However, the algorithm exploits primal degeneracy to perform the algebraic computations with a smaller *deficient* basis.

Our main contribution in this work is to introduce the dual counterpart of the positive edge criterion to reduce the effects of degeneracy in the dual simplex. This new pricing rule is supported by a description of the concept of dual compatibility that highlights the relationship with the primal case.

Compared with the work by Towhidi et al. [24], this article has a stronger emphasis on the theoretical study of the properties of the compatible variables. We study the stability of the set of compatible variables to justify that it does not need to be updated after each simplex pivot. To improve the implementation of Towhidi et al., we then determine a criterion to update the set of dual compatible variables only when necessary. Preliminary tests support the efficiency of our approach.

Finally, we implement the criterion within CLP and perform extensive computational tests to demonstrate its strong potential when solving degenerate instances. Our analysis of the results highlights the characteristic that has the greatest impact on the performance of the algorithm, thus suggesting how parameters could easily be set for an efficient adaptive use of the pricing criterion.

The article is organized as follows. In Section 2, we analyze dual compatibility and describe an efficient stochastic test to identify the dual compatible variables. We introduce the dual positive edge pricing criterion and discuss the implementation details in Section 3. The experimental results are presented and discussed in Section 4, and Section 5 provides concluding remarks.

[1] http://plato.asu.edu/ftp/lptestset/

2. Compatibility in the dual simplex

2.1. Notation

The theoretical developments focus on the LP in standard form:

$$(\mathrm{P}): \begin{cases} \min & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s. t.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{cases} \tag{1}$$

where $\boldsymbol{c}, \boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. The corresponding dual program is expressed in standard form:

$$(\mathrm{D}): \begin{cases} \min & \boldsymbol{b}^T \boldsymbol{y} \\ \text{s. t.} & \boldsymbol{A}^T \boldsymbol{y} + \boldsymbol{d} = \boldsymbol{c} \\ & \boldsymbol{y} \in \mathbb{R}^m, \boldsymbol{d} \geq \boldsymbol{0}, \end{cases} \tag{2}$$

where $\boldsymbol{d} \in \mathbb{R}^n$. We make the usual assumption that $\boldsymbol{A}$ is of full rank.

For any subset $\mathcal{J} \subseteq \{1, ..., n\}$ of column indices, the submatrix of $\boldsymbol{A}$ with columns indexed by $\mathcal{J}$ is denoted $\boldsymbol{A}_{\mathcal{J}}$, and $\boldsymbol{x}_{\mathcal{J}}$ is the subvector of variables indexed by $\mathcal{J}$. The vectors of all ones and all zeros with dimensions dictated by the context are denoted $\mathbf{1}$ and $\mathbf{0}$. For $p \in \mathbb{N} \setminus \{0\}$, $j \leq p$, the j^{th} vector of the canonical basis of $\mathbb{R}^p$ is denoted $\boldsymbol{e}_j^p$.

A basis $\mathcal{B}$ is an ordered set of m variable indices such that $\boldsymbol{A}_{\mathcal{B}}$ is nonsingular. The ordered set $\mathcal{N}$ indexes the remaining nonbasic variables. We then define the corresponding solution vector $\bar{\boldsymbol{b}} = \boldsymbol{A}_{\mathcal{B}}^{-1} \boldsymbol{b}$, reduced cost vector $\bar{\boldsymbol{c}} = \boldsymbol{c} - \boldsymbol{c}_{\mathcal{B}}^T \boldsymbol{A}_{\mathcal{B}}^{-1} \boldsymbol{A} \geq \mathbf{0}$, and simplex tableau $\bar{\boldsymbol{A}} = \boldsymbol{A}_{\mathcal{B}}^{-1} \boldsymbol{A}$. The basis $\mathcal{B}$ is feasible if and only if $\bar{\boldsymbol{b}} \geq \mathbf{0}$, and it is dual feasible if and only if $\bar{\boldsymbol{c}} \geq \mathbf{0}$.

For conciseness, the vector spaces spanned respectively by the columns and the rows of a matrix are referred to as the column and row spaces. The row space of a matrix $\boldsymbol{M}$ is the column space of $\boldsymbol{M}^T$.

Finally, the operator $\langle \cdot | \cdot \rangle$ refers to the scalar product.

2.2. A definition of dual compatibility

Generalizing the work done by Elhallaoui et al. [7] for set partitioning problems, Omer [19] has introduced the concept of compatibility in the algorithmic context of the primal simplex. The compatible variables are those that can be pivoted into the basis without impacting the values of the degenerate variables. The most general definition of compatibility is given for the primal case in [20].

Definition 1 (Compatibility). *Let $\mathcal{J}$ be a set of variable indices. A variable x_j, $j \in \{1, \ldots, n\}$, is compatible with $\mathcal{J}$ if the corresponding column $\boldsymbol{A}_j$ is in the column space of $\boldsymbol{A}_{\mathcal{J}}$.*

In this article, we are interested only in the variables that are compatible with the set $\mathcal{S}$ of variables that are strictly within their bounds. As a consequence, the variables compatible with $\mathcal{S}$ are simply referred to as "compatible."

To study the dual case, we first apply Definition 1 to the dual problem (D). Assuming that a feasible solution $(\boldsymbol{y}, \boldsymbol{d})$ of (D) is available, let $\mathcal{P}$ and $\overline{\mathcal{P}}$ denote the set of positive and zero variables of $\boldsymbol{d}$, i.e., $\boldsymbol{d}_{\mathcal{P}} > \mathbf{0}$ and $\boldsymbol{d}_{\overline{\mathcal{P}}} = \mathbf{0}$. In this solution, the variables strictly within their bounds are those in $\boldsymbol{y}$ and $\boldsymbol{d}_{\mathcal{P}}$. A direct application of Definition 1 shows that the variables in $\boldsymbol{y}$ and $\boldsymbol{d}_{\mathcal{P}}$ are compatible, so we are mostly interested in identifying the compatible variables of $\boldsymbol{d}_{\overline{\mathcal{P}}}$. Since (D) involves only unbounded ($\boldsymbol{y}$) and slack variables ($\boldsymbol{d}$), the proposition below states another characterization of the compatible variables for this particular model. In the remainder of this article, we use p and $\overline{p} = n - p$ to denote the cardinality of $\mathcal{P}$ and $\overline{\mathcal{P}}$ respectively.

Proposition 1. *Let j_i be the index of the i^{th} variable of $\overline{\mathcal{P}}$. Variable d_{j_i} is compatible if and only if $\boldsymbol{e}_i^{\overline{p}}$ is in the row space of $\boldsymbol{A}_{\overline{\mathcal{P}}}$.*

Proof. The constraint matrix of (D) is $(\boldsymbol{A}^T, \boldsymbol{I})$ where $\boldsymbol{I}$ is the identity matrix of $\mathbb{R}^n$. The variables strictly within their bounds are those of $\boldsymbol{y}$ and $\boldsymbol{d}_{\mathcal{P}}$, so the variable d_{j_i} is compatible if and only if the j_i^{th} column of $\boldsymbol{I}$ is in the column space of $(\boldsymbol{A}^T\ \boldsymbol{I}_{\mathcal{P}})$. This means that there exists a nonzero $(\boldsymbol{\alpha}_y, \boldsymbol{\alpha}_{\mathcal{P}}) \in \mathbb{R}^m \times \mathbb{R}^p$ such that

$$\begin{aligned}(\boldsymbol{A}_{\mathcal{P}})^T \boldsymbol{\alpha}_y + \boldsymbol{\alpha}_{\mathcal{P}} &= \boldsymbol{0} \\ (\boldsymbol{A}_{\overline{\mathcal{P}}})^T \boldsymbol{\alpha}_y &= \boldsymbol{e}_i^{\overline{p}}.\end{aligned}$$

Since $\boldsymbol{\alpha}_{\mathcal{P}}$ can always be set to satisfy the first equation, we consider only the second one. This equation has a nonzero solution if and only if $\boldsymbol{e}_i^{\overline{p}}$ is in the column space of $(\boldsymbol{A}_{\overline{\mathcal{P}}})^T$ or, equivalently, in the row space of $\boldsymbol{A}_{\overline{\mathcal{P}}}$. □

In the context of a dual simplex solution, it is more natural to assume that a dual feasible basis $\mathcal{B}$ of (P) is available, i.e., $\bar{\boldsymbol{c}}_{\mathcal{N}} \geq \boldsymbol{0}$, together with $\boldsymbol{x}_{\mathcal{B}} = \bar{\boldsymbol{b}}$ and $\boldsymbol{x}_{\mathcal{N}} = \boldsymbol{0}$. The dual degenerate variables are the nonbasic variables x_j, $j \in \mathcal{N}$, such that $\bar{c}_j = 0$. In this case, the correspondence between the dual and primal solutions can be stated with respect to the sets $\mathcal{P}$ and $\overline{\mathcal{P}}$.

Proposition 2. *A feasible solution of* (D) *is built by setting* $\boldsymbol{y}^T = \boldsymbol{c}_{\mathcal{B}}^T \boldsymbol{A}_{\mathcal{B}}^{-1}$ *and* $\boldsymbol{d} = \bar{\boldsymbol{c}}$. *Since* $\boldsymbol{d}_{\mathcal{P}} > \boldsymbol{0}$ *and* $\boldsymbol{d}_{\overline{\mathcal{P}}} = \boldsymbol{0}$, *we have* $\bar{\boldsymbol{c}}_{\mathcal{P}} > \boldsymbol{0}$ *and* $\bar{\boldsymbol{c}}_{\overline{\mathcal{P}}} = \boldsymbol{0}$. *The dual degenerate variables of* (P) *are thus associated with the degenerate variables of* (D).

Based on this correspondence, we use the characterization in Proposition 1 to define the dual compatible variables.

Definition 2 (Dual compatibility)**.** *With the notation of Proposition 1, the variable* x_{j_i}, $j_i \in \overline{\mathcal{P}}$, *is* dual compatible *if and only if* $\boldsymbol{e}_i^{\overline{p}}$ *is in the row space of* $\boldsymbol{A}_{\overline{\mathcal{P}}}$.

We may highlight the analogy between dual and primal compatibility by assuming that the columns of $\mathcal{A}$ are rearranged so that $\boldsymbol{A} = (\boldsymbol{A}_{\overline{\mathcal{P}}} \boldsymbol{A}_{\mathcal{P}})$. In that case, the variable x_j, $j \in \{1, \ldots, m\}$, is primal feasible if and only if $\boldsymbol{A}_j$ is in the column space of $\boldsymbol{A}_{\mathcal{S}}$, and it is dual compatible if and only if $\boldsymbol{e}_j^{\overline{p}}$ is in the row space of $\boldsymbol{A}_{\overline{\mathcal{P}}}$. Note that $\boldsymbol{A}_{\mathcal{S}}$ contains the columns of the primal nondegenerate variables, whereas $\boldsymbol{A}_{\overline{\mathcal{P}}}$ contains the columns of the dual degenerate variables.

Remark. *On a conceptual level, one important difference between primal and dual compatibility is that a basic solution is needed only in the dual case. If a solution of* (P) *is available, but no basis is known, the link with the dual program cannot be made. The consequence is that dual compatibility cannot be defined independently of the algorithm that is used to solve* (P). *In contrast, primal compatibility is exploited in [20] to derive a general decomposition scheme that finds an improvement in the objective value at each step. The algorithm considers the compatible and incompatible variables independently at each iteration to compute an improvement direction. The dual counterpart of this approach would require adaptations.*

The goal of this work is the development of a pivot selection rule for the dual simplex. We are thus interested in an algebraic characterization of the dual compatible variables that exploits knowledge of a basis associated with the dual feasible solution. For a basic solution, the reduced costs of the basic variables are all equal to zero, and the nonbasic variables with a zero reduced cost are the dual degenerate variables. The set $\overline{\mathcal{P}}$ is thus composed of the basic and the dual degenerate variables. Denoting by $\mathcal{Z} = \overline{\mathcal{P}} \cap \mathcal{N}$ the set of dual degenerate variables, we have $\overline{\mathcal{P}} = \mathcal{B} \cup \mathcal{Z}$.

Proposition 3. *The dual degenerate variables are not dual compatible, and the* i^{th} *basic variable is dual compatible if and only if* $(\bar{\boldsymbol{A}}_{\mathcal{Z}})^T \boldsymbol{e}_i^m = \boldsymbol{0}$.

Proof. The elements of $\overline{\mathcal{P}}$ can be ordered so that the first m elements correspond to the basic variables and the remaining $\overline{p} - m$ elements correspond to the dual degenerate variables. Let x_{j_i} be the i^{th} variable of $\overline{\mathcal{P}}$. Based on Definition 2, x_{j_i} is dual compatible if and only if there exists $\boldsymbol{\alpha} \in \mathbb{R}^m \neq \boldsymbol{0}$ such that

$$(\boldsymbol{A}_{\overline{\mathcal{P}}})^T \boldsymbol{\alpha} = \boldsymbol{e}_i^{\overline{p}} \Leftrightarrow \begin{pmatrix} (\boldsymbol{A}_{\mathcal{B}})^T \boldsymbol{\alpha} \\ (\boldsymbol{A}_{\mathcal{Z}})^T \boldsymbol{\alpha} \end{pmatrix} = \boldsymbol{e}_i^{\overline{p}}.$$

If x_{j_i} is dual degenerate, i.e., $i \in \{m+1, \ldots, \overline{p}\}$, the condition above implies that $(\boldsymbol{A}_{\mathcal{B}})^T \boldsymbol{\alpha} = \boldsymbol{0}$. Since the basic matrix is nonsingular, $\boldsymbol{\alpha} = \boldsymbol{0}$, so x_{j_i} is not dual degenerate.

If x_{j_i} is basic, i.e., $i \in \{1, \dots, m\}$,

$$\begin{pmatrix} (\boldsymbol{A}_{\mathcal{B}})^T \boldsymbol{\alpha} \\ (\boldsymbol{A}_{\mathcal{Z}})^T \boldsymbol{\alpha} \end{pmatrix} = \boldsymbol{e}_i^{\overline{p}} \Leftrightarrow (\boldsymbol{A}_{\mathcal{B}})^T \boldsymbol{\alpha} = \boldsymbol{e}_i^m \text{ and } (\boldsymbol{A}_{\mathcal{Z}})^T \boldsymbol{\alpha} = \boldsymbol{0}$$
$$\Leftrightarrow (\boldsymbol{A}_{\mathcal{Z}})^T (\boldsymbol{A}_{\mathcal{B}}^{-1})^T \boldsymbol{e}_i^m = \boldsymbol{0}.$$

With the more compact notation $\bar{\boldsymbol{A}} = \boldsymbol{A}_{\mathcal{B}}^{-1}\boldsymbol{A}$, x_{j_i} is thus dual compatible if and only if $(\bar{\boldsymbol{A}}_{\mathcal{Z}})^T \boldsymbol{e}_i^m = \boldsymbol{0}$. □

2.3. Identifying the dual compatible variables

In the dual simplex, the pricing criterion selects a pivot row i corresponding to an infeasible basic variable, i.e., $\bar{b}_i < 0$. During the ratio test, the entering variable is then chosen as

$$\operatorname*{argmin}_{j \in \mathcal{N}} \left\{ \frac{\bar{c}_j}{\bar{a}_{ij}} : \bar{a}_{ij} > 0 \right\}.$$

Assuming that the pivot row corresponds to a dual compatible variable, Proposition 3 implies that the dual degenerate variables will not be considered during the ratio test. As a consequence, a nondegenerate pivot is always performed when a dual compatible variable is selected to leave the basis. This suggests that we could speed up the algorithm by prioritizing dual compatible variables in the pricing step. To this end, the dual compatible variables need to be identified efficiently. The computational cost would be too high if the matrix $\boldsymbol{A}_{\mathcal{B}}^{-1}\boldsymbol{A}_{\mathcal{Z}}$ had to be computed at each iteration. We thus adapt the positive edge test that Raymond et al. [23] developed for the primal case.

Theorem 1 (Dual positive edge test). *Let x_{j_i} be the i^{th} basic variable. Let $\boldsymbol{v}$ be a vector of $\overline{p} - m$ continuous random variables. If x_{j_i} is dual compatible then $\langle (\bar{\boldsymbol{A}}_{\mathcal{Z}})^T \boldsymbol{e}_i^m | \boldsymbol{v} \rangle = 0$; otherwise, there is a zero probability that $\langle (\bar{\boldsymbol{A}}_{\mathcal{Z}})^T \boldsymbol{e}_i^m | \boldsymbol{v} \rangle = 0$.*

Proof. Proposition 3 states that x_{j_i} is dual compatible if and only if

$$(\bar{\boldsymbol{A}}_{\mathcal{Z}})^T \boldsymbol{e}_i^m = \boldsymbol{0},$$

so $\langle (\bar{\boldsymbol{A}}_{\mathcal{Z}})^T \boldsymbol{e}_i^m | \boldsymbol{v} \rangle = 0$. Otherwise, $\langle (\bar{\boldsymbol{A}}_{\mathcal{Z}})^T \boldsymbol{e}_i^m | \boldsymbol{v} \rangle$ is a continuous random variable. The probability that it takes a particular value is then zero. □

This theorem leads to a practical stochastic test for the identification of the dual compatible variables. Given a vector $\boldsymbol{v} \in \mathbb{R}^{\overline{p}-m}$ sampled from a continuous random variable, the i^{th} basic variable is dual compatible with a zero probability of error if and only if the i^{th} element of $\bar{\boldsymbol{A}}_{\mathcal{Z}}\boldsymbol{v}$ is equal to zero. We first compute

$$\boldsymbol{\alpha} = \boldsymbol{A}_{\mathcal{Z}}\boldsymbol{v}. \tag{3}$$

The vector $\boldsymbol{\rho} = \bar{\boldsymbol{A}}_{\mathcal{Z}}\boldsymbol{v}$ is then obtained by solving the system

$$\boldsymbol{A}_{\mathcal{B}}\boldsymbol{\rho} = \boldsymbol{\alpha}. \tag{4}$$

This highlights that the algebraic operations are similar to those involved in the computation of the reduced costs of the variables of $\mathcal{Z}$.

Remark. *In practice, limited floating-point precision makes it impossible to simulate continuous random variables, so the probability of identifying an incompatible variable as dual compatible cannot be zero. Raymond et al. [23] show that a well-chosen discrete random law can lead to a negligible probability of error. Moreover, our intent is to use the stochastic test in the pricing criterion, so the worst possible consequence of such an error is that the dual simplex performs an unexpected degenerate pivot.*

The dual compatible variables may be identified relatively cheaply, but the idea of focusing on these variables may be productive only if there are enough of them to consider in the pricing step. Intuitively, the number of dual compatible variables should decrease when the number of columns of $\bar{\boldsymbol{A}}_{\mathcal{Z}}$, i.e., the number of dual degenerate variables, increases. The following proposition states more precisely the relationship between the number of dual degenerate and dual compatible variables.

Proposition 4. *The number of dual compatible variables is less than or equal to* $m - \operatorname{rank} \boldsymbol{A}_{\mathcal{Z}}$.

Proof. Let C be the set of basic rows corresponding to the dual compatible variables. For all $i \in C$, $(\bar{\boldsymbol{A}}_{\mathcal{Z}})^T \boldsymbol{e}_i^m = (\boldsymbol{A}_{\mathcal{Z}})^T (\boldsymbol{A}_{\mathcal{B}}^{-1})^T \boldsymbol{e}_i^m = \boldsymbol{0}$, so

$$\operatorname{Span}(\{(\boldsymbol{A}_{\mathcal{B}}^{-1})^T \boldsymbol{e}_i^m : i \in C\}) \subset \operatorname{Ker}(\boldsymbol{A}_{\mathcal{Z}}^T).$$

Since $\boldsymbol{A}_{\mathcal{B}}$ is nonsingular, the dimension of $\operatorname{Span}(\{(\boldsymbol{A}_{\mathcal{B}}^{-1})^T \boldsymbol{e}_i^m : i \in C\})$ is equal to the cardinality of C, which proves the proposition. □

It is possible to build simple examples of dual degenerate solutions in which there are no dual compatible variables, or the number of dual compatible variables is equal to $m - \operatorname{rank} \boldsymbol{A}_{\mathcal{Z}}$. As a consequence, Proposition 4 gives the best theoretical bound on the number of dual compatible variables. In practice however, the number of dual compatible variables is expected to be positively correlated with $m - \operatorname{Card}(\mathcal{Z})$.

3. Implementation of the dual positive edge criterion

3.1. A two-dimensional pricing criterion

The theoretical developments of Section 2 suggest that the dual simplex could benefit from a pricing step that prioritizes the dual compatible variables. However, the various steepest-edge criteria compared in [8] have proved their efficiency, and they tend to reduce the effects of degeneracy [13]. For these reasons, we choose to implement the dual positive edge (`PE`) as a two-dimensional criterion, similarly to the implementation of Towhidi et al. [24] for the primal simplex.

`PE` selects a dual compatible variable only when it is not a bad choice with regards to the reference pricing criterion. To be more specific, all the well-known pricing criteria may be seen as normalized criteria. This means that they compute a vector of m positive weights $\boldsymbol{w}$, and they select a pivot row i such that

$$i \in \operatorname{argmin}\left\{\frac{\bar{b}_k}{w_k} : k = 1, \ldots, m\right\}.$$

In the Dantzig criterion, the weights are simply set to 1. Assuming that the current solution is not optimal, i.e., $\bar{\boldsymbol{b}} \not\geq \boldsymbol{0}$, the two-dimensional selection rule is summarized by Algorithm 1. The two-dimensional criterion relies on a parameter $0 \leq \psi \leq 1$, which corresponds to the level of priority that is given to the dual compatible variables. For instance, if $\psi = 0$ we always select a dual compatible variable when one is available, whereas dual compatibility is not considered if $\psi = 1$. In the rest of this article, C denotes the set of row indices corresponding to the dual compatible variables.

Algorithm 1: Two-dimensional dual positive edge pricing criterion

Input: The current solution and weight vectors $\bar{\boldsymbol{b}}$ and $\boldsymbol{w}$.
The set of dual compatible variables C.
Output: The pivot row i.

$i^{\min} \in \operatorname{argmin}\left\{\bar{b}_k/w_k : 1 \leq k \leq m\right\}$; $b^{\min} \leftarrow \min\left\{\bar{b}_k/w_k : 1 \leq k \leq m\right\}$;
$i_C^{\min} \in \operatorname{argmin}\left\{\bar{b}_k/w_k : k \in C\right\}$; $b_C^{\min} \leftarrow \min\left\{\bar{b}_k/w_k : k \in C\right\}$;
if $b_C^{\min} \leq \psi \times b^{\min}$ **then**
  $i \leftarrow i_C^{\min}$;
else
  $i \leftarrow i^{\min}$;

3.2. Practical identification of the dual compatible variables

Most efficient simplex implementations apply the Gilbert–Peierls method [10] to take advantage of sparsity in the right-hand side of the linear systems. Since $\boldsymbol{v}$ is randomly generated, the right-hand side of the system (4) is 100% dense with probability 1. As a consequence, the identification of the dual compatible variables will generally take more time than the computation of the reduced cost vector. If these operations are performed at each simplex iteration, the overhead will be too large to be compensated for by the expected reduction in the number of iterations.

The two propositions below suggest that the set of dual compatible variables should not change dramatically after each simplex pivot, thus justifying less frequent updates of $\mathcal{C}$. In the statements and proofs of these propositions, we denote by the symbol $'$ the sets and values corresponding to the state of the solution after the pivot. Moreover, x_l is the leaving variable selected with the pricing criterion, the entering variable x_e is deduced from the ratio test, and i is the pivot row associated with x_l.

Proposition 5. *If a degenerate pivot is performed, then $\mathcal{C}' = \mathcal{C}$.*

Proof. After a pivot, the reduced costs and the simplex tableau can be updated with the following formulas:

$$\bar{c}'_j \leftarrow \bar{c}_j - \bar{c}_e \times \frac{\bar{a}_{ij}}{\bar{a}_{ie}}, \quad 1 \leq j \leq n \tag{5}$$

$$\bar{a}'_{kj} \leftarrow \bar{a}_{kj} - \bar{a}_{ke} \times \frac{\bar{a}_{ij}}{\bar{a}_{ie}}, \quad 1 \leq k \leq m, 1 \leq j \leq n. \tag{6}$$

For a degenerate pivot, the entering variable must be dual degenerate, i.e., $\bar{c}_e = 0$. Equation (5) shows that the vector of reduced costs is unchanged. The set of dual degenerate variables is then updated by removing x_e and adding x_l, i.e., $\mathcal{Z}' = \mathcal{Z} \setminus \{e\} \cup \{l\}$.

Assume that $k \in \mathcal{C}$. Since $e \in \mathcal{Z}$, Proposition 3 implies that $\bar{a}_{ke} = 0$. Equation (6) then shows that the k^{th} row of $\bar{\boldsymbol{A}}$ is not modified by the pivot, so $k \in \mathcal{C}'$.

Assume that $k \notin \mathcal{C}$. If $\bar{a}_{ke} = 0$, the k^{th} row of $\bar{\boldsymbol{A}}$ is not modified, so $k \notin \mathcal{C}'$. If $\bar{a}_{ke} \neq 0$,

$$\bar{a}'_{kl} = 0 - \bar{a}_{ke} \times \frac{1}{\bar{a}_{ie}} \neq 0.$$

Since $l \in \mathcal{Z}'$, Proposition 3 implies that $k \notin \mathcal{C}'$. □

Proposition 6. *Assume that a nondegenerate pivot is performed. If a dual compatible variable is selected to leave the basis, then $\mathcal{C}' \supset \mathcal{C}$. Moreover, if no additional degeneracy is created by the pivot, i.e., $\mathcal{Z}' \subset \mathcal{Z}$, then $\mathcal{C}' = \mathcal{C}$.*

Proof. If x_l is dual compatible, i.e., $i \in \mathcal{C}$, then $\bar{a}_{ij} = 0$ for all $j \in \mathcal{Z}$. Equation (5) shows that for all $j \in \mathcal{Z}$, $\bar{c}'_j = \bar{c}_j = 0$, so $\mathcal{Z}' \supset \mathcal{Z}$. By (6), $\bar{\boldsymbol{A}}'_{\mathcal{Z}} = \bar{\boldsymbol{A}}_{\mathcal{Z}}$, so $\mathcal{C}' \supset \mathcal{C}$.

If we also assume that $\mathcal{Z}' \subset \mathcal{Z}$, then $\mathcal{Z}' = \mathcal{Z}$. Since $\bar{\boldsymbol{A}}'_{\mathcal{Z}} = \bar{\boldsymbol{A}}_{\mathcal{Z}}$, we have $\mathcal{C}' = \mathcal{C}$. □

A nonsystematic update of $\mathcal{C}$ may cause two errors. First, dual compatible variables may not be identified as such. Opportunities to perform nondegenerate pivots may then be missed, but this will not cause a bad decision in the pivot selection. Second, an incompatible variable may be tagged as compatible. This situation is more troublesome, since the two-dimensional criterion could overlook the best choice and select a leaving variable that does not even guarantee an improvement in the objective value. Propositions 5 and 6 show that the two types of errors can happen when a nondegenerate pivot is done with an incompatible leaving variable. Moreover, the second type of error can happen if a nondegenerate pivot creates additional degeneracy. Since the stability of $\mathcal{C}$ is not guaranteed after all the possible pivots, it is necessary to periodically update the set of dual compatible variables to ensure that PE remains efficient.

Towhidi et al. [24] update $\mathcal{C}$ when they observe a significant change in the number of dual degenerate variables. This mechanism responds to the appearance of a probable cause of error. We prefer to intervene when actual difficulties arise, i.e., when degenerate pivots are performed after we select variables of $\mathcal{C}$ as the leaving variables. More precisely, let $dgn_{\mathcal{C}}$ be the percentage of degenerate pivots when a variable of $\mathcal{C}$ leaves the basis, and dgn the overall percentage of degenerate pivots. An update of $\mathcal{C}$ is performed as soon as $dgn_{\mathcal{C}}$ becomes higher than a small percentage ρ of dgn. We also set a minimum number of iterations between two consecutive updates, $it_{\min}$, to prevent the

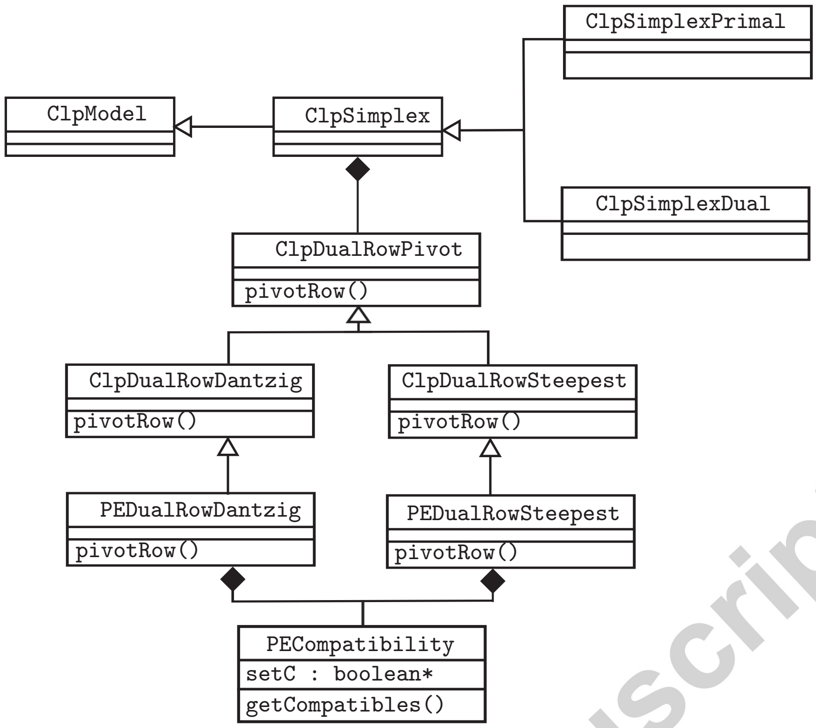

Figure 1: Partial UML class diagram for CLP with the dual positive edge criterion

updates from slowing the dual simplex excessively. Finally, an update may be needed even if dgn_C remains low. For instance, if there is no dual compatible variable in the initial solution, dgn_C will be equal to zero until an update of C is performed. As a consequence, an update is also performed after $it_{\max}$ consecutive iterations without an update. In our tests we set $\rho = 0.2$, $it_{\min} = 50$, and $it_{\max} = 1000$.

Remark. *The stability of the set of compatible variables stated in Propositions 5 and 6 is actually essential for the concept of compatibility. As already noted, the recent interest in compatibility is motivated by the opportunity for a nondegenerate pivot. However, the compatible variables are not the only ones with this property. Nevertheless, compatibility was used even before the development of a fast approach for the identification of the compatible variables. For instance, the improved primal simplex (IPS) of Elhallaoui et al. [6] is based on compatibility, and it was developed before the positive edge criterion. In the first versions of the algorithm, IPS computed the columns of the simplex tableau to identify the compatible variables. Despite this large computational effort, good results were obtained in practice because it is not necessary to identify the compatible variables at each iteration.*

3.3. Implementation within COIN-OR LP solver

PE modifies the reference pricing criterion to include the two-dimensional selection. The pricing step is in the core of the simplex algorithm, so it is impossible to implement the PE criterion without accessing the source code of an implementation of the dual simplex. CLP is an open-source LP solver written in C++ that includes a dual simplex. Moreover, Koberstein [15] reports good performance for the dual simplex of CLP. CLP thus appears to be appropriate for testing the performance of PE.

The partial unified model language (UML) class diagram in Figure 1 focuses on the portion of CLP that implements the dual simplex and specifies the classes added for the positive edge criterion. `ClpDualRowPivot` is the base class for dual simplex pivots in CLP. `ClpDualRowDantzig` implements the classical Dantzig pricing criterion, and `ClpDualRowSteepest` implements the dual steepest edge criterion. The selection of the leaving variable is done in the `pivotRow()` method. We thus define the two-dimensional criterion corresponding to `ClpDualRowDantzig` and `ClpDualRowSteepest` by deriving two subclasses `PEDualRowDantzig` and `PEDualRowSteepest` that implement modified versions of `pivotRow()`. The method `getCompatibles()` of the class `PECompatibility` computes the set of dual compatible variables and stores the corresponding indices in the attribute `setC`.

4. Computational tests

In Section 4.2, we provide the numerical test results for the internal CLP implementation over a relatively large benchmark described in Section 4.1. We perform our experiments using computers with Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz processors.

During the tests, we introduce PE within the dual simplex of CLP equipped with the dual steepest edge (DSE) and the dual Dantzig (DD) pricing rules. Based on preliminary results, we set the priority level ψ to 0.1 in DD and to 0.4 in DSE. The reason for choosing different values is that the DSE pricing rule is designed to avoid degenerate pivots and small steps. If we give a high priority to the dual compatible variables, PE may actually work against DSE.

4.1. Description of the benchmark

We run the tests on Mittelmann's LP test set[2], which is used in [24] to validate the primal positive edge. To extend the benchmark, we also consider a set of LPs listed in the LinLIB, which contains more than 500 instances collected from four well-known libraries including Netlib[3], the Kennington problems[4], and the BPMPD benchmark[5]. The LinLIB is organized into five size categories depending on the number of nonzero elements in the constraint matrix. To control the number of instances, we restricted our tests to the most difficult instances. We thus used the instances with more than 50000 nonzero elements that took more than 5 s to solve with the dual steepest edge simplex of CLP.

The dimensions of the 62 selected instances are given in Table 1. CLP systematically adds one slack variable for each constraint; we count only the decision variables. For a better insight into the difficulty of solving each instance, we also provide the numbers of pivots (it_{CLP}) and computational times (t_{CLP}) for the dual simplex of CLP with the DSE and DD pricing rules. The time limit for the CLP algorithm was set to 10 hours. A "t" in the it_{CLP} and t_{CLP} columns indicates that the instance could not be solved within the time limit. We do not report results for the cont11, nug20, nug30, and L1_d10_40 instances, because none of the tested algorithms (with or without PE) is able to solve them within the time limit.

Table 1: Benchmark: Dimensions and solution with CLP

	problem dimensions			dual steepest edge		dual Dantzig	
Instance	constraints	variables	nonzeros	it_{CLP}	t_{CLP}(s)	it_{CLP}	t_{CLP}(s)
co9	10789	14851	101578	10997	6.1	33435	14.4
cont1	160792	40398	399990	54789	285.4	52100	877.4
cont4	160792	40398	398398	53970	340.8	49433	798.8
cq9	9278	13778	88897	12092	6.0	17178	6.2
dano3mip	3202	13873	79655	44872	39.1	605250	519.8
dbic1	43200	183235	1038761	119210	805.2	258383	1410.4
ds-big	1042	174997	4623442	57906	540.1	1050090	11646.8
ex3sta1	17443	8156	59419	9433	11.0	20764	65.1
fast0507	507	63009	409349	3934	6.3	92682	185.4
fome11	12142	24460	71264	34909	20.5	2338627	2279.5
fome12	24284	48920	142528	71263	48.4	4469849	7086.2
fome13	48568	97840	285056	142249	116.4	9019866	25349.2
fome20	33874	105728	230200	23471	23.5	2389110	4671.0
fome21	67748	211456	460400	55028	73.0	6565852	30108.1

Continued on next page

[2] http://plato.asu.edu/ftp/lptestset/
[3] http://www.netlib.org/lp/data/
[4] http://www.netlib.org/lp/data/kennington/
[5] http://www.sztaki.hu/~meszaros/public_ftp/lptestset/

Table 1: continued from previous page

	Problem dimensions			Dual steepest edge		Dual Dantzig	
Instance	constraints	variables	nonzeros	it_{CLP}	$t_{\mathrm{CLP}}(\mathrm{s})$	it_{CLP}	$t_{\mathrm{CLP}}(\mathrm{s})$
fxm3-16	41340	64162	370839	51370	8.0	47175	10.2
gen2	1121	3264	81855	6121	7.5	188463	242.8
gen4	1537	4297	107102	6951	11.6	12502	19.8
ken-18	105127	154699	358171	39023	10.0	242539	780.9
l30	2701	15380	51169	9222	7.3	28864	16.8
Linf-520c	93326	69004	566193	235660	3590.2	219731	3082.5
lp22	2958	13434	65560	18155	13.1	677737	827.0
lpl1	39951	125000	381259	29542	29.4	7182991	30547.5
mod2	34774	31728	165129	37669	54.1	1112986	2187.8
model10	4400	15447	149000	68598	60.5	282136	375.9
model11	7056	18288	55859	29835	21.3	433141	270.9
model5	1888	11360	89483	41312	9.8	57810	19.9
nemswrld	7138	27174	190907	25225	28.4	148047	162.5
neos	479119	36786	1047675	60291	213.5	757154	4415.8
neos1	131581	1892	468009	42946	199.2	4036208	20030.1
neos2	132568	1560	552519	122160	670.8	3409450	17265.0
neos3	512209	6624	1542816	76272	2371.4	1168894	19392.7
ns1644855	40698	30200	2110696	65557	360.9	201275	524.9
ns1687037	50622	43749	1406739	60235	680.7	t	t
ns1688926	32768	16587	1712128	166980	3282.7	95670	221.9
nug08-3rd	19728	20448	139008	28692	249.9	130852	2173.9
nug15	6330	22275	94950	t	t	t	t
osa-30	4350	100024	600138	2793	6.2	3657	8.3
osa-60	10280	232966	1397793	5653	31.5	9167	54.2
pds-030	49944	154998	337144	61593	106.2	t	t
pds-040	66844	212859	462128	108308	276.2	t	t
pds-050	83060	270095	585114	130909	405.4	t	t
pds-060	99431	329643	712779	170076	569.6	t	t
pds-070	114944	382311	825771	313306	1538.9	t	t
pds-080	129181	426278	919524	332561	1617.5	t	t
pds-090	142823	466671	1005359	408780	2265.9	t	t
pds-100	156243	505360	1086785	365961	1960.5	t	t
pilot87	2030	4883	73152	19053	14.6	39222	28.5
rail2586	2586	920683	8008776	26982	845.1	t	t
rail4284	4284	1092610	11279748	56628	2281.8	t	t
scfxm1-2r-256	37980	57714	213159	35794	7.0	56325	11.9
self	960	7364	1148845	3557	14.4	20714	78.2
south31	18425	35421	111498	17831	11.5	19268	9.4
stat96v1	5995	197472	588798	57502	212.6	137086	361.9
stat96v4	3173	62212	490472	295189	612.8	1518448	4093.5
stormG2-1000	528185	1259121	3341696	474765	1539.0	882336	4781.4
stormG2-125	66185	157496	418321	59922	19.6	86210	35.0
stp3dlp	159488	204880	662128	98587	389.2	3139094	33021.5
t0331-4l	664	46915	430982	7539	8.1	16502	18.5
ulevimin	6590	44605	162206	25452	28.6	148208	134.1
watson1	201155	383927	1052028	204105	268.7	184023	216.1
watson2	352013	671861	1841028	321584	1071.7	303880	605.0
world	34506	32734	164470	43686	66.6	1953499	4134.7

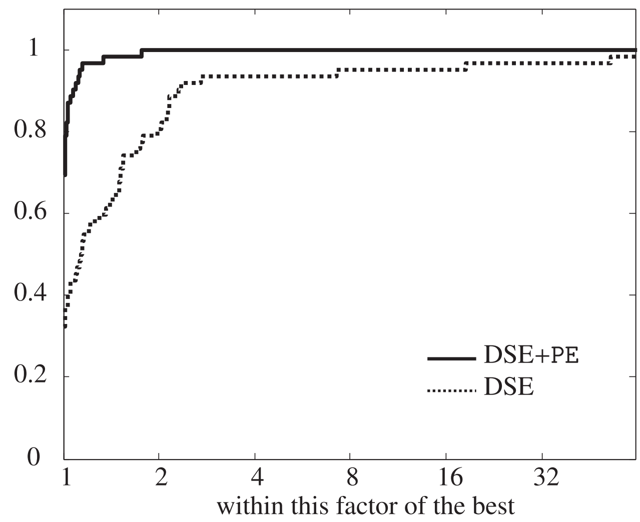

Figure 2: Performance profile over solution time with the dual steepest edge criterion

4.2. Results

For a global representation of the results, we draw the performance profile of the dual simplex with and without PE. This visualization technique is very efficient for large benchmarks, because it highlights the global behavior rather than isolated cases. Moreover, performance profiles remain clear when the algorithms fail to solve some instances or when the solutions involve large gaps between the represented values. A detailed description of performance profiling is given in [5].

The DSE and DD simplex algorithms of CLP are compared with the corresponding algorithms equipped with PE in the two performance profiles in Figures 2 and 3. The two profiles clearly indicate that the two-dimensional selection improves the DD and DSE simplex of CLP. For instance, the solution times with the DSE and DD simplex are improved by a factor larger than two for more than 20% and 30% respectively of the instances. Moreover, PE improves the solution times of both dual simplex algorithms significantly (by more than 15%) for more than 50% of the instances. In contrast, PE has a significant negative impact for less than 5% of the instances, and the resulting increase in the solution time is below a factor of two (see Figure 2).

We also provide the average improvement in the number of pivots (*it*) and the average speedup (*t*). The speedup is computed as the ratio $t_{\texttt{CLP}}/t_{\texttt{PE}}$, where $t_{\texttt{PE}}$ and $t_{\texttt{CLP}}$ are the solution times of the CLP's DSE simplex respectively with and without the positive edge criterion. A similar computation provides the improvement in the number of pivots. The first row of Table 2 records the geometric means of these values, averaged over the instances that could be solved within the time limit. The number f of instances that each method failed to solve within the time limit is also displayed. The results show that PE improved the solution time of the DSE simplex by an average factor of 1.42, and that of the DD simplex by an average factor of 1.95. A comparison of the two columns $\frac{t_{\texttt{CLP}}}{t_{\texttt{PE}}}$ and $\frac{it_{\texttt{CLP}}}{it_{\texttt{PE}}}$ confirms that the speedup is mostly caused by a reduction in the number of pivots. Moreover, the two-dimensional pivot rule allowed us to solve 50% of the instances that could not be solved within the time limit with the DD simplex. We do not display the time that PE spent in the update of the dual compatible variables, because it never took more than 1% of the total solution time.

The performance profiles show that PE has a negative impact for only a small fraction of the instances. However, the new pricing rule does not always lead to a significant improvement. In the last three rows of Table 2, we investigate two parameters that may impact the performance of PE. One factor that has a major impact is the number of compatible variables. For instance, if no basic variable is compatible or if they all are, PE does not modify the selected pivot row. Similarly, PE should not have a strong impact for the instances that exhibit a small or very large number of dual compatible variables. In the second row of Table 2, the results focus on the instances for which the average number of dual compatible variables is between 1% and 99% of the number of rows. Another important statistic is the percentage of degenerate pivots. The PE algorithm is specifically designed to mitigate the negative effects of degeneracy, so the

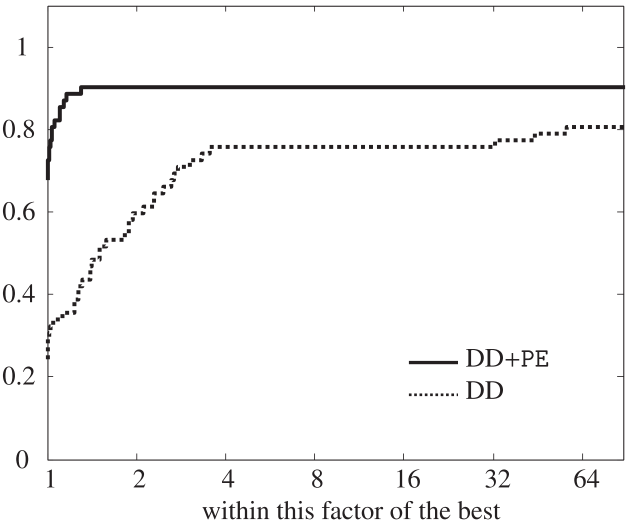

Figure 3: Performance profile over solution time with the dual Dantzig criterion

Table 2: Average improvement with the dual positive edge algorithms

	dual Dantzig				dual steepest edge			
	instances	$\frac{it_{\text{CLP}}}{it_{\text{PE}}}$	$\frac{t_{\text{CLP}}}{t_{\text{PE}}}$	$f_{\text{CLP}} - f_{\text{PE}}$	instances	$\frac{it_{\text{CLP}}}{it_{\text{PE}}}$	$\frac{t_{\text{CLP}}}{t_{\text{PE}}}$	$f_{\text{CLP}} - f_{\text{PE}}$
every instance	62	1.69	1.95	12 – 6	62	1.24	1.42	5 – 4
C1: $0.01m \leq \lvert C\rvert \leq 0.99m$	48	1.95	2.30	12 – 6	49	1.31	1.56	1 – 0
C2: degen. pivots > 20%	44	1.98	2.33	12 – 6	39	1.40	1.61	1 – 0
C1 and C2	36	2.36	2.88	11 – 5	31	1.49	1.76	1 – 0

instances that have a low level of degeneracy should not require PE. The third row restricts the study to the instances for which more than 20% of the pivots were degenerate. Finally, the last row displays the performance of PE for the instances that satisfy both the above conditions. For each algorithm, the column “instances” contains the number of instances that satisfy the associated condition.

The results in Table 2 confirm that PE is more useful when the solution performs a large number of degenerate pivots and an intermediate ratio of compatible variables is detected. Specifically, the last row of the table shows that at least 50% of the instances satisfy both conditions. When we consider only these instances, the average speedup increases from 1.42 to 1.76 and from 1.95 to 2.88 for DSE and DD respectively. The number of pivots follows the same trend.

Finally, we perform additional tests to study the impact of not updating the set of dual compatible variables at each iteration (see Section 3.2). To this end, we modify the PE algorithm to update the dual compatible variables before every pivot. The results confirm that every pivot involving a variable of C is nondegenerate when the update is performed at each iteration. However, the solution time increases by 60% on average due to the computational time spent in the update of C. In contrast, the number of pivots remains about the same, so if C is updated at each iteration the overall solution time of the DSE simplex is on average higher with PE than without PE. This highlights that exploiting the relative stability of the set of dual compatible variables is essential for an efficient implementation of PE.

5. Discussion and conclusions

The first contribution of this article is in the extension of the concept of compatibility to the dual simplex. Our theoretical developments establish the link between primal and dual compatibility and provide an algebraic characterization of compatibility in the context of the dual simplex. The main property of the dual compatible variables is that they give rise to a nondegenerate pivot when selected to leave the basis. Since the characterization is similar to that for the primal case, we are able to derive a stochastic test similar to that developed in [23] for a fast identification of the dual compatible variables. We apply this test in a two-dimensional selection rule (`PE`). `PE` can be combined with any dual pricing criterion to prioritize the compatible variables during the selection of the pivot row.

We implemented `PE` in the dual simplex of the open-source `CLP` solver. A key feature of the implementation is the frequency of the update of the set of compatible variables C. Since C is not altered by degenerate pivots and by most nondegenerate pivots on dual compatible variables, it is not necessary to update it at each iteration. We do the update only when a significant number of degenerate pivots are done with pivot rows in C.

The computational tests were performed on a large benchmark including 62 instances from well-known linear programming benchmarks. We focus on two classical pricing rules: the dual Dantzig criterion (DD) and the dual steepest edge (DSE). On average, `PE` improves the solution times of DSE by 42% and those of DD by 95% for the instances that can be solved in less than ten hours. For the other instances, `PE` allows us to solve one out of five instances with DSE and six out of sixteen with DD. Moreover, when `PE` is not able to improve the pricing rules, it has a small negative impact or no impact at all on the solution time.

The effect of `PE` is stronger over DD than it is over DSE, because the latter pricing rule is efficient in reducing the negative impact of degeneracy. This leaves less room for improvement in the two-dimensional selection rule when DSE is used.

A comparison of the number of simplex iterations shows that the speedup is mostly caused by a reduction in the number of pivots. However, the speedup is greater than the improvement factor in the number of pivots for both DSE and DD. This trend is also individually respected by the instances. This suggests that it is favorable to the computational efficiency of the simplex to stay in the subspace of the compatible variables.

Finally, the results also confirmed that `PE` is more efficient when at least 1% and at most 99% of the basic variables are dual compatible and when the dual simplex performs more than 20% degenerate pivots. This shows that `PE` should be implemented as an adaptive strategy that is triggered only when these criteria are met. It also suggests that a better version of the selection rule could set the priority level ψ according to the percentage of compatible variables.

It would be interesting to investigate combining `PE` with the dynamic pricing criterion of Klotz [14]. This criterion can identify variables that will lead to a degenerate pivot. It could be combined with the concept of compatibility to derive a three-dimensional selection rule that gives a low priority to the variables that will lead to a degenerate pivot and a high priority to the compatible variables.

Finally, IPS [6] is a primal decomposition scheme that dynamically removes constraints from degenerate problems. At each major iteration of the algorithm, IPS uses the primal simplex to solve a reduced problem containing the nondegenerate and the compatible variables. In a recent work [21], the authors described a revised version of IPS that identifies the compatible variables with the (primal) positive edge test. This algorithm performs much better on large instances. The dual counterpart of the revised IPS would be able to take advantage of dual degeneracy to reduce the size of the problem and make quick progress toward optimality. This should lead to better performance for some degenerate problems.

References

[1] Banciu, M., Cochran, J. J., Cox, L. A., Keskinocak, P., Kharoufeh, J. P., Smith, J. C., 2010. Dual Simplex. John Wiley & Sons, Inc.

[2] Benichou, M., Gauthier, J. M., Hentges, G., Ribiere, G., 1977. The efficient solution of large-scale linear programming problems: Some algorithmic techniques and computational results. Mathematical Programming 13 (1), 280–322.

[3] Bixby, R. E., 2002. Solving real-world linear programs: A decade and more of progress. Operations Research 50, 3–15.

[4] Dantzig, G. B., 1955. The general simplex method for minimizing a linear form under inequality constraints. Pacific Journal of Mathematics 5, 183–195.

[5] Dolan, E. D., Moré, J. J., 2002. Benchmarking optimization software with performance profiles. Mathematical Programming 91 (2), 201–213.

[6] Elhallaoui, I., Metrane, A., Desaulniers, G., Soumis, F., 2011. An improved primal simplex algorithm for degenerate linear programs. INFORMS Journal on Computing 23 (4), 569–577.

[7] Elhallaoui, I., Villeneuve, D., Soumis, F., Desaulniers, G., 2005. Dynamic aggregation of set-partitioning constraints in column generation. Operations Research 53 (4), 632–645.
[8] Forrest, J. J. H., Goldfarb, D., 1992. Steepest-edge simplex algorithms for linear programming. Mathematical Programming 57, 341–371.
[9] Fourer, R., 1994. Notes on the dual simplex method.
[10] Gilbert, J., Peierls, T., 1988. Sparse partial pivoting in time proportional to arithmetic operations. SIAM Journal on Scientific and Statistical Computing 9 (5), 862–874.
[11] Gill, P. E., Murray, W., Saunders, M. A., Wright, M. H., 1989. A practical anti-cycling procedure for linearly constrained optimization. Mathematical Programming 45 (1–3), 437–474.
[12] Greenberg, H. J., 1978. Design and Implementation of Optimization Software. Sijthoff & Noordhoff, Ch. Pivot Selection Tactics, pp. 109–142.
[13] Hattersley, B., Wilson, J., 1988. A dual approach to primal degeneracy. Mathematical Programming 42 (1), 135–145.
[14] Klotz, E. S., 1988. Dynamic pricing criteria in linear programming. Tech. Rep. SOL 88-15, Stanford University.
[15] Koberstein, A., 2008. Progress in the dual simplex algorithm for solving large scale LP problems: Techniques for a fast and stable implementation. Computational Optimization and Applications 41 (2), 185–204.
[16] Lemke, C. E., 1954. The dual method of solving the linear programming problem. Naval Research Logistics Quarterly 1 (1), 36–47.
[17] Lougee-Heimer, R., 2003. The Common Optimization INterface for Operations Research: Promoting open-source software in the operations research community. IBM Journal of Research and Development 47 (1), 57–66.
[18] Maros, I., 2003. A generalized dual phase-2 simplex algorithm. European Journal of Operational Research 149 (1), 1–16.
[19] Omer, J., 2006. Méthode de réduction dynamique de contraintes dans un programme linéaire. Master's thesis, École Polytechnique de Montréal.
[20] Omer, J., Rosat, S., Raymond, V., Soumis, F., 2014. Improved primal simplex: A more general theoretical framework and an extended experimental analysis, Les Cahiers du GERAD G-2014-13, GERAD, Montreal, Quebec, Canada.
[21] Omer, J., Soumis, F., 2014. A linear programming decomposition focusing on the span of the nondegenerate columns, Les Cahiers du GERAD G-2014-41, GERAD, Montreal, Quebec, Canada.
[22] Pan, P.-Q., 2005. A revised dual projective pivot algorithm for linear programming. SIAM Journal on Optimization 16 (1), 49–68.
[23] Raymond, V., Soumis, F., Metrane, A., Desrosiers, J., 2010. Positive edge: A pricing criterion for the identification of non-degenerate simplex pivots, les Cahiers du GERAD G-2010-61, GERAD, Montreal, Quebec, Canada.
[24] Towhidi, M., Desrosiers, J., Soumis, F., 2014. The positive edge criterion within COIN-OR's CLP. Computers & Operations Research 49, 41–46.