Submitted to *Operations Research*
manuscript

Vector Space Decomposition for Linear Programs

Jean Bertrand Gauthier, Jacques Desrosiers
HEC Montréal and GERAD, Montréal, Canada H3T 2A7
jean-bertrand.gauthier@hec.ca, jacques.desrosiers@hec.ca

Marco E. Lübbecke
RWTH Aachen University, Operations Research
Kackertstraße 7, D-52072 Aachen, Germany
marco.luebbecke@rwth-aachen.de

This paper describes a vector space decomposition algorithmic framework for linear programming guided by dual feasibility considerations. The resolution process moves from one basic solution to the next according to an exchange mechanism which is defined by a direction and a post-evaluated step size. The core component of this direction is obtained via the smallest reduced cost that can be achieved upon dividing the set of dual variables in two subsets: one being fixed whiles the other being optimized. From a primal perspective, such a pricing problem selects a convex combination of variables entering the basis. The direction is uniquely completed by identifying *a posteriori* affected variables, if any. The degenerate status of the current solution can be exploited in specific variants.

This unified framework is presented in a generic format from which different variants can be extracted. These include several well known algorithms. The most known special case is the Primal Simplex algorithm where all dual variables are fixed: this results in the choice of a single entering variable, indeed a myopic strategy commonly leading to degenerate pivots. At the other extreme, we find the strongly polynomial Minimum Mean Cycle-Canceling algorithm for the capacitated network flow problem for which all dual variables are optimized at every iteration. Somewhere in between these two extremes lies another special case, the Improved Primal Simplex algorithm for which one fixes the dual variables associated with the nondegenerate basic variables and optimizes the rest.

The two last variants both bestow a pricing problem providing necessary and sufficient optimality conditions. As a result, nondegenerate directions are also issued from these pricing steps. These directions move on the edges of the polyhedron for the latter while the former can identify interior directions by combining edge directions. Other properties are derived on different variants of this generic framework.

Key words: primal simplex algorithm; degeneracy; optimized reduced costs; combination of entering variables; positive edge rule; nondegenerate pivot algorithms; vector space.

Contents

1. Introduction

While Dantzig's Primal Simplex algorithm (PS) dates back to the summer of 1947 (Dantzig and Thapa 1997, p. xxvi), it still commands attention today. The race for shaving seconds off the resolution process started soon after its birth and indeed remains strong today. Although researchers keep breaking records, the original simplicity of the algorithm has been traded for it. In other words, it is hard to deny that PS has stood the test of time, but then again each one of its steps is invested by mathematical details which make an efficient implementation nontrivial. Think of the basis matrix factorization maintenance, partial pricing strategies or primal-dual exploits.

One of the most critical phenomenon encountered in practice is degeneracy. The latter introduces stalling in the resolution process which may even lead to cycling. An important survey on anti-cycling schemes and pivot-selection rules can be found in Terlaky and Zhang (1993) where a very large number of these are examined. In the words of Dantzig and Thapa (2003, p. 167), *whether any degeneracy avoiding scheme is needed in practice has never been settled. It has been observed, however, that even when there is no degeneracy, there is a high probability of near-degeneracy. This suggests that pivot-selection criteria should be designed to seek feasible solutions in directions away from degenerate and near-degenerate basic feasible solutions, or better yet, driven by dual feasibility considerations.*

The whole paragraph is the perfect embodiment of a rhetorical question. Although there is a lot of evidence that suggests accounting for degeneracy one way or another can be profitable, in no way does it positively asserts the need for such schemes. We therefore

acknowledge these observations and address their final remark whereby pivot-selection rules guided by dual information seem favorable.

Of course, at the end of the day, all known primal algorithms base their stopping criteria on the optimality conditions provided by dual variables. Since no two algorithms share the same iterative process, there must be different levels of exposure that can be harvested from the realm of dual feasibility. Klotz (1988) extensively studies oracle based pricing methods for linear programs in an effort to side step degeneracy. The author is keen to observe that a trade-off between the number of iterations and the computational cost of the pricing requires significant additional work even when comparing to Dantzig's pivot-selection rule. As unfortunate as this may seem, one of the most forthcoming idea is that of *variable screening*. By discarding variables from the pricing problem, one hopes to dramatically reduce the difficulty of solving it. While many of the proposed screening rules try to get a handle on the problem's structure and transfer the information to the pricing problem, we present a design for the latter that inherently and dynamically exploits the problem structure. Such concerns also already appeared within column generation, the extension of PS for linear programs with a large number of variables, see Lübbecke and Desrosiers (2005). Degeneracy in column generation has been dealt with using certain dual variable stabilization approaches, see for example du Merle et al. (1999) and Ben Amor et al. (2009) for a stabilized column generation framework and the many references therein.

In this paper, we propose an algorithmic framework which, given a feasible basic solution, *fixes the values of a subset of dual variables* and optimizes the remaining ones for finding the smallest reduced cost (until one reaches an optimal solution). It turns out that the dual formulation of this pricing problem selects a convex combination of variables entering the basis. The way to divide these two subsets of dual variables relies on the choice of a vector subspace basis and opens a wide spectrum of possibilities thus paving the way for the paper at hand. Since each dual variable is associated with a constraint, this division also amounts to a partition of the rows.

The most known special case, Dantzig's pivot-selection rule for PS, considers all dual variables fixed and selects a single entering variable, a myopic strategy commonly resulting in degenerate pivots. At the other extreme, when none of the dual variables are fixed, we stumble upon the Minimum Mean Cycle-Canceling algorithm (MMCC) devised for capacitated network flow problems and shown to be strongly polynomial by Goldberg and Tarjan (1989). Another remarkable variant is that defined by the Improved Primal Simplex algorithm (IPS) (Elhallaoui et al. 2011) which not only prevents degenerate pivots but also maintains the basic status of solutions. It thus stands out sharing some of the features of both extremes of the spectrum and, as such, guards the crossroad of the later. Last but not least, it also encapsulates the Dynamic Constraint Aggregation method (DCA) of Elhallaoui et al. (2005, 2008). The latter is specifically designed to overcome degeneracy in the context of solving the linear relaxation of set partitioning models by column generation. Although DCA is a precursor of IPS, its methodology slightly differs in that the row partition is deduced using a heuristic design. It works well because the row partition is in line with the purpose of set partitioning problems. The vector space decomposition

framework introduced in Gauthier et al. (2015) closes the theoretical gap between IPS and DCA.

The paper is organized as follows. Section 2 takes a close look at the essential components of the framework. Several concepts that partake (or not) in the resolution process of a linear program are examined such as nondegerate pivots, directions and the not yet defined *residual problem*. Each of these is presented in a separate manner whereas the last subsection ties everything together. Section 3 builds upon these ties and gives birth to the generic algorithm. In Section 4, we demonstrate a few properties, determine a family of algorithms with nondegenerate pivots at every iteration, and show that some directions are interior rather than along edges. Moreover, we examine well known special cases. Indeed, by using vector space decomposition, the paper at hand unifies within the same generic algorithm a variety of specialized ones for linear and network programs. We conclude in Section 5 with our contribution and some research perspectives.

2. The problem

Consider the linear program (LP) with lower and upper bounded variables:

$$\begin{aligned} z^\star := \min \quad & \mathbf{c}^\intercal \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \ [\boldsymbol{\pi}] \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned} \tag{1}$$

where $\mathbf{x}, \mathbf{c}, \mathbf{l}, \mathbf{u} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $m < n$. We assume that the matrix $\mathbf{A}$ is of full row rank and that LP (1) is feasible and bounded. The vector of dual variables $\boldsymbol{\pi} \in \mathbb{R}^m$ associated with the equality constraints appears within brackets on the right-hand side.

Notation. Vectors and matrices are written in bold face. We denote by $\mathbf{I}_r$ the $r \times r$ identity matrix and by $\mathbf{0}$ (resp. $\mathbf{1}$) a vector/matrix with all zeros (resp. ones) entries of contextually appropriate dimension. For an ordered subset $R \subseteq \{1, \ldots, m\}$ of row indices and an ordered subset $P \subseteq \{1, \ldots, n\}$ of column indices, we denote by $\mathbf{A}_{RP}$ the sub-matrix of $\mathbf{A}$ containing the rows and columns indexed by R and P, respectively. We further use standard linear programming notation like $\mathbf{A}_B\mathbf{x}_B$, the subset of basic columns of $\mathbf{A}$ indexed by B multiplied by the corresponding vector of basic variables $\mathbf{x}_B$. The set of nonbasic columns N is used analogously. The lower case notation is reserved for vectors and uses the same subset index rules. In particular, the matrix $\mathbf{A} := (\mathbf{a}_j)_{j \in \{1,\ldots,n\}}$ contains n column vectors.

In Section 2.1, we formulate the so-called *residual problem* which allows the construction of an oracle generating only strictly feasible directions in Section 2.2. The latter also provides two alternative primal and dual conditions characterizing optimality for linear programs. Finally, let us embark upon this generic algorithm in Section 2.3 with an arbitrary invertible matrix $\mathbf{T}$ whose utility becomes clear when we merge these two theories together.

2.1. The residual problem

When PS performs a nondegenerate pivot at iteration $k \geq 0$, it goes from vertex $\mathbf{x}^k$ represented by a nonoptimal basis to vertex $\mathbf{x}^{k+1}$ by moving along an edge (Dantzig and Thapa 2003, Theorem 1.7). The truth of the matter is that this movement is taken by construction of the entering/exiting variable mechanism. In the case of a degenerate pivot, the basis is modified but the geometrical solution vertex remains the same. In other words, the movement induced by the selected negative reduced cost entering variable leads outside the polyhedron and we do not move. By thinking of these possible movements as directions, the residual problem presented next provides a way to identify these, and many others, in a general framework.

It is common practice in developing network flow algorithms to use a *residual network* to improve upon some intermediate solution by identifying incremental flows, see Ahuja et al. (1993). In this paper, we do the same with linear programs and provide primal-dual optimality conditions on the residual problem.

Let $\mathbf{x}^k$ be a feasible solution to LP (1) at iteration $k \geq 0$ with cost value $z^k = \mathbf{c}^\intercal \mathbf{x}^k$. Perform a change of variables,

$$\mathbf{x} := \mathbf{x}^k + \mathbf{y} = \mathbf{x}^k + (\vec{\mathbf{y}} - \overleftarrow{\mathbf{y}}), \qquad \mathbf{0} \leq \vec{\mathbf{y}} \leq \vec{\mathbf{r}}^k,\ \mathbf{0} \leq \overleftarrow{\mathbf{y}} \leq \overleftarrow{\mathbf{r}}^k,\ \vec{\mathbf{y}}^\intercal \overleftarrow{\mathbf{y}} = 0, \tag{2}$$

where the forward and backward vectors $\vec{\mathbf{y}}, \overleftarrow{\mathbf{y}} \in \mathbb{R}^n_+$, whereas their residual upper bounds are given by $\vec{\mathbf{r}}^k := \mathbf{u} - \mathbf{x}^k$ and $\overleftarrow{\mathbf{r}}^k := \mathbf{x}^k - \mathbf{l}$. We define the *residual problem* $LP(\mathbf{x}^k)$ with respect to a given solution $\mathbf{x}^k$ as follows. Each variable $x_j,\ j \in \{1, \ldots, n\}$, in the original LP (1) is replaced by two oriented variables: the forward variable $\vec{y}_j$ of cost c_j represents the possible increase of x_j relatively to x^k_j while the backward variable $\overleftarrow{y}_j$ of cost $-c_j$ represents its possible decrease; moreover, only one can be used with a positive value, i.e., complementarity condition $\vec{y}_j \overleftarrow{y}_j = 0,\ \forall j \in \{1, \ldots, n\}$, see Figure 1.

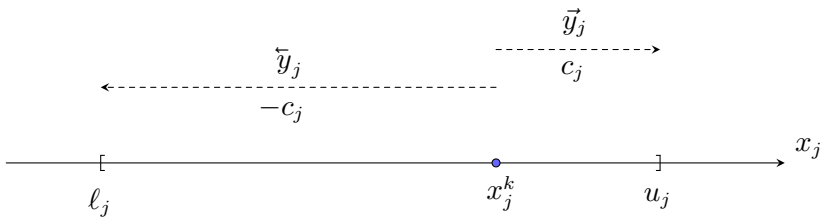

Figure 1 Forward and backward vectors for the residual problem

Equivalent to LP (1), a formulation for the residual problem $LP(\mathbf{x}^k)$ is as follows:

$$\begin{aligned} z^\star := z^k \ + \quad \min \quad & \mathbf{c}^\intercal(\vec{\mathbf{y}} - \overleftarrow{\mathbf{y}}) \\ \text{s.t.} \quad & \mathbf{A}(\vec{\mathbf{y}} - \overleftarrow{\mathbf{y}}) = \mathbf{0} \quad [\boldsymbol{\pi}] \\ & \mathbf{0} \leq \vec{\mathbf{y}} \leq \vec{\mathbf{r}}^k \\ & \mathbf{0} \leq \overleftarrow{\mathbf{y}} \leq \overleftarrow{\mathbf{r}}^k. \end{aligned} \tag{3}$$

It should come as no surprise that the residual problem $LP(\mathbf{x}^k)$ needs only be solved for y-variables with positive residual upper bounds, i.e., a variable bounded under and above by 0 is useless. While the residual problem is *unique*, let us nonetheless define the residual problem $RP(\mathbf{x}^k)$ where variables with null residual upper bounds are explicitly discarded. In order to achieve this, let $\mathbf{x}^k$ be represented by $(\mathbf{x}^k_F;\mathbf{x}^k_L;\mathbf{x}^k_U)$, where the three sub-vectors are defined according to the value of the variables they contain: $\mathbf{x}^k_L$ at their lower bounds, $\mathbf{x}^k_U$ at their upper bounds, and *free* variables $\mathbf{l}_F < \mathbf{x}^k_F < \mathbf{u}_F$ which can move below or above their current value. Let there be $f := |F|$ such free variables, $0 \le f \le n$. Observe that if $\mathbf{x}^k$ is basic then $0 \le f \le m$. Partition the matrix $\mathbf{A} = [\mathbf{A}_F, \mathbf{A}_L, \mathbf{A}_U]$ and the cost vector $\mathbf{c}^\intercal = [\mathbf{c}^\intercal_F, \mathbf{c}^\intercal_L, \mathbf{c}^\intercal_U]$ accordingly. Taking into account the column partition $\{F, L, U\}$, $RP(\mathbf{x}^k)$ becomes

$$\begin{array}{rlrcll} z^\star := \mathbf{c}^\intercal\mathbf{x}^k \; + & \min & \mathbf{c}^\intercal_F(\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F) & + & \mathbf{c}^\intercal_L\vec{\mathbf{y}}_L - \mathbf{c}^\intercal_U\overleftarrow{\mathbf{y}}_U & \\ & \text{s.t.} & \mathbf{A}_F(\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F) & + & \mathbf{A}_L\vec{\mathbf{y}}_L - \mathbf{A}_U\overleftarrow{\mathbf{y}}_U & = \; \mathbf{0} \quad [\boldsymbol{\pi}] \\ & & \vec{\mathbf{y}}_F \ge \mathbf{0},\ \overleftarrow{\mathbf{y}}_F \ge \mathbf{0}, & & \vec{\mathbf{y}}_L \ge \mathbf{0},\ \overleftarrow{\mathbf{y}}_U \ge \mathbf{0} & \\ & & \vec{\mathbf{y}}_F \le \vec{\mathbf{r}}^k_F,\ \overleftarrow{\mathbf{y}}_F \le \overleftarrow{\mathbf{r}}^k_F, & & \vec{\mathbf{y}}_L \le \vec{\mathbf{r}}^k_L,\ \overleftarrow{\mathbf{y}}_U \le \overleftarrow{\mathbf{r}}^k_U. & \end{array} \tag{4}$$

2.2. An oracle

From the residual problem formulation $LP(\mathbf{x}^k)$ in (3), it is obvious that any *direction* $\mathbf{y} = (\vec{\mathbf{y}} - \overleftarrow{\mathbf{y}}) \in \{\mathbf{A}(\vec{\mathbf{y}} - \overleftarrow{\mathbf{y}}) = \mathbf{0},\ \vec{\mathbf{y}}, \overleftarrow{\mathbf{y}} \ge \mathbf{0}\}$ is *feasible* for LP (1). Observe that a direction containing a variable with a null residual upper bound, that is, $\vec{y}_j > 0, j \in U$, or $\overleftarrow{y}_j > 0, j \in L$, necessarily leads to a degenerate pivot. By discarding all these variables with null residual upper bounds from the residual problem formulation $RP(\mathbf{x}^k)$ in (4), any *direction* $\mathbf{y} = \begin{bmatrix} (\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F) \\ \vec{\mathbf{y}}_L \\ -\overleftarrow{\mathbf{y}}_U \end{bmatrix}$ is *strictly feasible* for LP (1) regardless of the remaining residual upper bounds.

Consider the following domain definition:

$$\begin{array}{rcrcrcl} \mathbf{A}_F(\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F) & + & \mathbf{A}_L(\vec{\mathbf{y}}_L - \overleftarrow{\mathbf{y}}_L) & + & \mathbf{A}_U(\vec{\mathbf{y}}_U - \overleftarrow{\mathbf{y}}_U) = & \mathbf{0} \\ \vec{\mathbf{y}}_F \ge \mathbf{0}, \overleftarrow{\mathbf{y}}_F \ge \mathbf{0}, & & \vec{\mathbf{y}}_L \ge \mathbf{0}, \overleftarrow{\mathbf{y}}_L \ge \mathbf{0}, & & \vec{\mathbf{y}}_U \ge \mathbf{0}, \overleftarrow{\mathbf{y}}_U \ge \mathbf{0}. & & \end{array} \tag{5}$$

Discarding the residual uppers bounds gives rise to many possible constructions of this domain which can either contain feasible directions or only the strictly feasible directions, a subset of the former. Indeed, $\overleftarrow{\mathbf{y}}_L$ and $\vec{\mathbf{y}}_U$ that would otherwise be well-behaved in (3) can be solicited depending on whether or not the domain (5) considers their removal or not.

Observe that the domain (5) defines a cone where each extreme ray corresponds to a direction. Define a *weighted cycle* as a direction $\mathbf{y}$, in line with the content of (5), satisfying an additional *normalizing* constraint, say $\mathbf{1}^\intercal\vec{\mathbf{y}} + \mathbf{1}^\intercal\overleftarrow{\mathbf{y}} = 1$ or $\mathbf{1}^\intercal\vec{\mathbf{y}}_F + \mathbf{1}^\intercal\overleftarrow{\mathbf{y}}_F + \mathbf{1}^\intercal\vec{\mathbf{y}}_L + \mathbf{1}^\intercal\overleftarrow{\mathbf{y}}_U = 1$. The latter cuts the cone and provides a polyhedron in which each extreme point is a bijection of an extreme ray in the cone.

Necessary and sufficient optimality conditions. The reduced cost of variable x_j is

defined as $\bar{c}_j = c_j - \boldsymbol{\pi}^\intercal \mathbf{a}_j$ while those of variables $\vec{y}_j$ and $\overleftarrow{y}_j$ are respectively plus and minus $\bar{c}_j$. In addition to the complementary slackness optimality conditions on LP (1) based on the reduced cost of the original $\mathbf{x}$-variables (see Schrijver 1986), we provide two alternative conditions characterizing optimality for linear programs. Similar to properties derived on network flow problems, these are based on an *Augmenting Weighted Cycle Theorem* on $LP(\mathbf{x}^k)$ (3) which is itself derived from a *Decomposition Theorem* of any feasible solution to LP (1).

Proposition 1 (Gauthier et al. 2014a, Theorem 4) *A feasible solution $\mathbf{x}^k$ to LP (1) is optimal if and only if the following equivalent conditions are satisfied:*

Complementary slackness: $\exists \boldsymbol{\pi}$ *such that* $\bar{c}_j > 0 \Rightarrow j \in L$; $\bar{c}_j < 0 \Rightarrow j \in U$; $j \in F \Rightarrow \bar{c}_j = 0$.
Primal: $RP(\mathbf{x}^k)$ *contains no negative cost weighted cycle.*
Dual: $\exists \boldsymbol{\pi}$ *such that the reduced cost of every variable of* $RP(\mathbf{x}^k)$ *is nonnegative.*

As supporting evidence for the primal condition claim, let the variables contained in the weighted cycle ω be $W := \{j \in \{F, L, U\} \mid y_j = (\vec{y}_j - \overleftarrow{y}_j) \neq 0\}$ and define the weighted cycle cost as $c_\omega := \sum_{j \in W} c_j y_j$. Analogously to the cost and reduced cost of a network cycle being equal, the cost c_ω and reduced cost $\bar{c}_\omega$ are also the same. Indeed, a weighted cycle satisfies (5), hence $\sum_{j \in W} \mathbf{a}_j y_j = \mathbf{0}$ and $\bar{c}_\omega = \sum_{j \in W} (c_j - \boldsymbol{\pi}^\intercal \mathbf{a}_j) y_j = c_\omega - \boldsymbol{\pi}^\intercal \sum_{j \in W} \mathbf{a}_j y_j = c_\omega$. The dual condition claim is trivially verified using the duality principle.

Finally, an oracle for generating improving directions is derived automatically from the domain (5) when the normalizing constraint is enlisted along with any objective function that ensures a negative cost weighted cycle can be identified if any remains. Notice that an oracle respects the sufficient condition if and only if the domain corresponds to the strictly improving directions.

2.3. Linear algebra

Observe that applying the inverse of an arbitrary nonsingular matrix $\mathbf{T}$ on the equality constraints of LP (1) yields an equivalent system, i.e., $\bar{\mathbf{A}}\mathbf{x} = \bar{\mathbf{b}} \Leftrightarrow \mathbf{T}^{-1}\mathbf{A}\mathbf{x} = \mathbf{T}^{-1}\mathbf{b}$. With respect to linear programming, it also implies a transformed set of dual variables $\boldsymbol{\psi} = \mathbf{T}\boldsymbol{\pi}$ for any given optimal primal solution. We invite the reader not to rationalize on $\mathbf{T}$ for now and let the paper speak for itself.

Let us recall some notions of linear algebra. Let $\boldsymbol{\Lambda}_r$ be a set of r linearly independent column vectors of $\mathbb{R}^m$. The set $\mathbf{V}(\boldsymbol{\Lambda}_r)$ of the linear combinations of these vectors is the vector subspace of $\mathbb{R}^m$ spanned by subspace basis $\boldsymbol{\Lambda}_r$. Its dimension is r and any set of r linearly independent vectors of $\mathbf{V}(\boldsymbol{\Lambda}_r)$ establishes a vector subspace basis for it. Let $\boldsymbol{\Lambda}_r^\perp := \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{m-r} \end{bmatrix}$ be a vector subspace basis for the orthogonal subspace $\mathbf{V}(\boldsymbol{\Lambda}_r)^\perp$ of dimension $m-r$. Then, $\mathbf{T}_r := [\boldsymbol{\Lambda}_r, \boldsymbol{\Lambda}_r^\perp]$ is a basis of $\mathbb{R}^m$.

As the vector subspace basis $\boldsymbol{\Lambda}_r$ contains r linearly independent column vectors, it must be possible to identify r independent rows, say the row set $R \subseteq \{1, \ldots, m\}$. Let the remainder of these dependent rows be in the row set $S := \{1, \ldots, m\} \setminus R$. Hence, $\boldsymbol{\Lambda}_r = \begin{bmatrix} \boldsymbol{\Lambda}_R \\ \boldsymbol{\Lambda}_S \end{bmatrix}$ and an alternative subspace basis for $\mathbf{V}(\boldsymbol{\Lambda}_r)$ is $\begin{bmatrix} \mathbf{I}_r \\ \mathbf{M} \end{bmatrix}$, where $\mathbf{M} := \boldsymbol{\Lambda}_S \boldsymbol{\Lambda}_R^{-1}$. Assuming the

rows of the matrix $\mathbf{A}$ follow the ordered set $\{R, S\}$, a generic version of the matrix $\mathbf{T}_r$, its inverse $\mathbf{T}_r^{-1}$ and $\bar{\mathbf{A}}$ can therefore be written as

$$\mathbf{T}_r = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{M} & \mathbf{I}_{m-r} \end{bmatrix}, \qquad \mathbf{T}_r^{-1} = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ -\mathbf{M} & \mathbf{I}_{m-r} \end{bmatrix}, \qquad \bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_R \\ \mathbf{A}_S - \mathbf{M}\mathbf{A}_R \end{bmatrix}. \tag{6}$$

The inverse matrix $\mathbf{T}_r^{-1}$ is itself composed of four sub-matrices where r and $m-r$ are respectively the sizes of R and S. The matrix $\mathbf{M}$ contains information about the *assumed/observed* linear dependence of the constraints of LP (1). The linear dependence of the constraints is interpreted with respect to the elements contained in some arbitrary basic solution.

Observe that by Gaussian elimination, any column vector $\mathbf{a} \in \mathbb{R}^m$ that belongs to the linear span of $\left[\begin{smallmatrix} \mathbf{I}_r \\ \mathbf{M} \end{smallmatrix}\right]$ contains only zeroes in the row set S, that is, $\bar{\mathbf{a}}_{Sj} := \mathbf{a}_{Sj} - \mathbf{M}\mathbf{a}_{Rj} = \mathbf{0}$ (see Proposition 2). Such is the *structure* that $\mathbf{T}_r^{-1}$ aims to make emerge. Its purpose is ultimately to *transform* the matrix $\mathbf{A}$ such that a four section partition surfaces.

The basis $\mathbf{A}_B$ in PS is a subspace basis of $\mathbb{R}^m$ which spans any and all entering variables. Since the linear combination scalars can take any sign, every column of $\mathbf{A}_B$ is implicitly expected to have freedom to move in either direction. This *could* unfortunately be proven false when the pivot exercise arrives. This possibility can only arise when a nonbasic variable is defined by a linear combination containing at least one degenerate basic variable. Indeed, the associated direction to such an entering variable *might* include a y-variable with a residual upper bound of 0. Named after its purpose, the transformation matrix $\mathbf{T}_r^{-1}$ breaks down the subspace of $\mathbb{R}^m$ into two smaller orthogonal subspaces. One can *associate* the subspace bases $\mathbf{\Lambda}_r$ and $\mathbf{\Lambda}_r^{\perp}$ with free and degenerate variables respectively. Whether this association is true or not dictates the vulnerability to degeneracy as illustrated by the different variations of the proposed generic algorithm.

3. Vector space decomposition framework

In this section, we look at the essential components of the proposed framework. The algorithm relies on an oracle to iterate. The latter is dynamically updated with respect to the values of the current solution. As such, we already stated the residual problem which gives a great deal of insight to the oracle's solutions. We also define a row/column partition, based on the transformation matrix, that generates the content of the oracle. In a nutshell, the sections obtained from this partition communicate with each other in the same way a master/sub-problem would. In practice, we capitalize on the partition by treating its content like a Dantzig-Wolfe decomposition, see Dantzig and Wolfe (1960).

The generic algorithm is decomposed into six main steps aside from the initialization. Figure 2 provides an overview of these steps, while each subsection details their content.

Initialization. It all starts with iteration counter $k = 0$ and some basic feasible solution $\mathbf{x}^0$ with column partition $\{F, L, U\}$. The construction of the residual problem $LP(\mathbf{x}^k)$ (3) calls for a change of variables: $\mathbf{x} := \mathbf{x}^k + \mathbf{y} = \mathbf{x}^k + (\vec{\mathbf{y}} - \overleftarrow{\mathbf{y}})$, $\vec{\mathbf{y}}, \overleftarrow{\mathbf{y}} \geq \mathbf{0}$, with appropriate residual upper bounds $\vec{\mathbf{r}}^k$, $\overleftarrow{\mathbf{r}}^k$ as well as complementarity condition $\vec{\mathbf{y}}^\intercal \overleftarrow{\mathbf{y}} = 0$.

Initialization: $k=0$, $\mathbf{x}^k$ basic, column partitions $\{B,N\}$ and $\{F,L,U\}$, $LP(\mathbf{x}^k)$.
1 **Subspace basis** $\mathbf{\Lambda}_r$ induces row partition $\{R,S\}$ of $\mathbf{A}$, matrix $\mathbf{T}_r$ and inverse $\mathbf{T}_r^{-1}$.
2 **Compatibility** with $\mathbf{\Lambda}_r$ induces column partition $\{P,Q\}$ of $\mathbf{A}$.
3 **Pricing step** optimizes minimum reduced cost μ given $\boldsymbol{\psi}_R^\intercal = \mathbf{c}_P^\intercal \mathbf{A}_{RP}^{-1}$.
4 If $\mu=0$, *stop with an optimal solution*, else build **Direction** $\mathbf{y} = \begin{bmatrix} \mathbf{y}_P \\ \mathbf{y}_Q \end{bmatrix}$.
5 **Exchange mechanism** computes step size ρ and $\mathbf{x}^k + \rho\mathbf{y}$.
6 **Data structures update** for $\mathbf{x}^{k+1}$, $\{B,N\}$ and $\{F,L,U\}$, $LP(\mathbf{x}^{k+1})$; goto Step 1.

Figure 2 Generic algorithm

3.1. Subspace basis $\mathbf{\Lambda}_r$

The selection of the subspace basis in Step 1 gives the flavor of the algorithm. The subspace basis is typically inspired by the free variable columns $\mathbf{A}_F$. Consider $\mathbf{\Lambda}_r = \begin{bmatrix} \mathbf{\Lambda}_R \\ \mathbf{\Lambda}_S \end{bmatrix}$, a set of r linearly independent vector in $\mathbb{R}^m$, where $\mathbf{\Lambda}_R$ is a nonsingular matrix of size r. If not already given, a row partition $\{R,S\}$ of $\mathbf{\Lambda}_r$ can be determined by solving a *restricted* Primal Simplex phase I problem over columns of $\mathbf{\Lambda}_r$ and those of the identity matrix $\mathbf{I}_m$ with appropriate dimensional vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$:

$$\min \ \{\mathbf{1}^\intercal\boldsymbol{\theta} \mid \mathbf{\Lambda}_r\boldsymbol{\lambda} + \mathbf{I}_m\boldsymbol{\theta} = \mathbf{\Lambda}_r\mathbf{1},\ \boldsymbol{\lambda} \geq \mathbf{0},\ \boldsymbol{\theta} \geq \mathbf{0}\}. \tag{7}$$

Any basic solution to (7) comprises r positive $\boldsymbol{\lambda}$-variables at one and $(m-r)$ $\boldsymbol{\theta}$-variables at zero. The row partition $\{R,S\}$ of $\mathbf{\Lambda}_r$ not only induces the row partition $\begin{bmatrix} \mathbf{A}_R \\ \mathbf{A}_S \end{bmatrix}$ of $\mathbf{A}$ but also the construction of the vector basis $\mathbf{T}_r$, the inverse $\mathbf{T}_r^{-1}$, and $\bar{\mathbf{A}} := \mathbf{T}^{-1}\mathbf{A}$, as already given in (6).

3.2. Compatibility with $\mathbf{\Lambda}_r$

Recall from Section 2.3 that the purpose of $\mathbf{T}_r^{-1}$ is to induce a partition in the matrix $\mathbf{A}$, that is, to identify column vectors with null entries in row set S. Column vectors that feature this structure are called compatible. The truth is there is something more fundamental about it. The linear dependence assumed in the system of equality of LP (1) means that an improving column $\mathbf{a} \in \mathbb{R}^m$ also honors that dependence. This leads us the formal definition of compatibility.

Definition 1 (Gauthier et al. 2015, Proposition 3) *A vector* $\mathbf{a} \in \mathbb{R}^m$ *(and the associated variable, if any) is* compatible with $\mathbf{\Lambda}_r$ *if and only if it belongs to* $\mathbf{V}(\mathbf{\Lambda}_r)$*, the linear span of* $\mathbf{\Lambda}_r$*.*

Verifying compatibility can be done using any and all methods available from the linear algebra arsenal. Some are more efficient than others depending on the content of matrix $\mathbf{A}$, the most probing known cases being the network and set partitioning problems which easily permit the verification of the definition. The following proposition allows to check the compatibility status by observing the content of the transformed rows of column $\mathbf{a}$.

Proposition 2 *A vector* $\mathbf{a} \in R^m$ *(and the associated variable, if any) is* compatible with $\mathbf{\Lambda}_r$ *if and only if* $\bar{\mathbf{a}}_S := \mathbf{a}_S - \mathbf{M}\mathbf{a}_R = \mathbf{0}$*, where* $\mathbf{M} = \mathbf{\Lambda}_S \mathbf{\Lambda}_R^{-1}$.

Proof. Assume that $\mathbf{a}$ is compatible with respect to Definition 1 such that there exists some vector $\boldsymbol{\alpha} \in \mathbb{R}^r$ for which $\begin{bmatrix}\mathbf{a}_R \\ \mathbf{a}_S\end{bmatrix} = \begin{bmatrix}\mathbf{\Lambda}_R \\ \mathbf{\Lambda}_S\end{bmatrix}\boldsymbol{\alpha}$. Since $\mathbf{\Lambda}_R$ is invertible, $\boldsymbol{\alpha} = \mathbf{\Lambda}_R^{-1}\mathbf{a}_R$ is uniquely defined. It is then trivial to observe that $\mathbf{a}_S = \mathbf{\Lambda}_S\mathbf{\Lambda}_R^{-1}\mathbf{a}_R$ if and only if Definition 1 holds. Observe that row set R can be arbitrarily chosen as long as it corresponds to r independent rows. □

For the record, *Positive Edge* (PE) is an alternative rule described in Raymond et al. (2010), Towhidi et al. (2014), and Gauthier et al. (2015) which uses a stochastic argument to reduce the matrix multiplication computational penalty. The rule uses a random vector $\mathbf{v} \neq \mathbf{0} \in \mathbb{R}^{m-r}$ and *declares* a column vector $\mathbf{a}$ compatible if $\mathbf{v}^\intercal[-\mathbf{M} \quad \mathbf{I}_{m-r}]\mathbf{a} = \mathbf{v}^\intercal\mathbf{T}_S^{-1}\mathbf{a} = \mathbf{w}^\intercal\mathbf{a} = 0$, where the premultiplication of vector $\mathbf{w} \in \mathbb{R}^m$ procures the computational savings. Testing for compatibility over basis $\mathbf{A}_B$ can be done in $O(m^2)$.

As it stands, there are exactly r basic columns that verify the compatibility definition. Let these be included in index set $P := \{j \in B \mid \mathbf{a}_j \in \mathbf{V}(\mathbf{\Lambda}_r)\}$. Let $Q := \{1, \ldots, n\} \setminus P$ contain the rest of the variables. This identification induces the column partition of matrix $\mathbf{A} = [\mathbf{A}_P, \mathbf{A}_Q]$. Altogether, we have the row/column partition $\mathbf{A} = \begin{bmatrix}\mathbf{A}_{RP} & \mathbf{A}_{RQ} \\ \mathbf{A}_{SP} & \mathbf{A}_{SQ}\end{bmatrix}$, where nonsingular matrix $\mathbf{A}_{RP}$ is called the *working basis*. Applying $\mathbf{T}_r^{-1}$ on $\mathbf{A}$ yields

$$\bar{\mathbf{A}} = \begin{bmatrix}\mathbf{I}_r & \mathbf{0} \\ -\mathbf{M} & \mathbf{I}_{m-r}\end{bmatrix}\begin{bmatrix}\mathbf{A}_{RP} & \mathbf{A}_{RQ} \\ \mathbf{A}_{SP} & \mathbf{A}_{SQ}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{RP} & \mathbf{A}_{RQ} \\ \mathbf{0} & \bar{\mathbf{A}}_{SQ}\end{bmatrix}. \tag{8}$$

3.3. Pricing step

The reformulation of $LP(\mathbf{x}^k)$ (3) is immediate from $\bar{\mathbf{A}}$. In the spirit of the residual problem paradigm, we are particularly interested in variables with positive residual upper bounds. Crossovers between the column partitions $\{P, Q\}$, $\{B, N\}$ and $\{F, L, U\}$ permit a tractable notation. For instance, $B_L := B \cap L$ denotes the index set of the degenerate basic variables at their lower bounds. Note that in any basic solution, all free variables are indeed nondegenerate basic variables, hence $B_F = F$ and $B = F \cup B_L \cup B_U$ whereas $N = N_L \cup N_U$.

Going back on topic for the positive residual upper bounds, the index set Q contains $Q_F \cup Q_L \cup Q_U$ whereas both forward $\vec{\mathbf{y}}_{Q_U}$-variables and backward $\overleftarrow{\mathbf{y}}_{Q_L}$-variables can be discarded. Respectively define the index sets for the remaining variables as $\vec{Q} := Q_F \cup Q_L$ and $\overleftarrow{Q} := Q_F \cup Q_U$ for the forward and backward $\mathbf{y}_Q$-variables.

$$
\begin{array}{rlrcll}
z^\star := z^k \; + & \min & \mathbf{c}_P^\intercal(\vec{\mathbf{y}}_P - \overleftarrow{\mathbf{y}}_P) & + & \mathbf{c}_{\vec{Q}}^\intercal \vec{\mathbf{y}}_{\vec{Q}} - \mathbf{c}_{\overleftarrow{Q}}^\intercal \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} & \\
& \text{s.t.} & \mathbf{A}_{RP}(\vec{\mathbf{y}}_P - \overleftarrow{\mathbf{y}}_P) & + & \mathbf{A}_{R\vec{Q}} \vec{\mathbf{y}}_{\vec{Q}} - \mathbf{A}_{R\overleftarrow{Q}} \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} & = \mathbf{0} \quad [\boldsymbol{\psi}_R] \\
& & \vec{\mathbf{y}}_P \le \vec{\mathbf{r}}_P^k,\ \overleftarrow{\mathbf{y}}_P \le \overleftarrow{\mathbf{r}}_P^k, & & \vec{\mathbf{y}}_{\vec{Q}} \le \vec{\mathbf{r}}_{\vec{Q}}^k,\ \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} \le \overleftarrow{\mathbf{r}}_{\overleftarrow{Q}}^k & \\
& & \vec{\mathbf{y}}_P \ge \mathbf{0},\ \overleftarrow{\mathbf{y}}_P \ge \mathbf{0} & & & \\
& & & & \bar{\mathbf{A}}_{S\vec{Q}} \vec{\mathbf{y}}_{\vec{Q}} - \bar{\mathbf{A}}_{S\overleftarrow{Q}} \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} & = \mathbf{0} \quad [\boldsymbol{\psi}_S] \\
& & & & \vec{\mathbf{y}}_{\vec{Q}} \ge \mathbf{0},\ \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} \ge \mathbf{0}. &
\end{array}
\tag{9}
$$

The row set R acts like in PS. Since the working basis $\mathbf{A}_{RP}$ is a subspace basis of $\mathbb{R}^r$, any column of Q that enters the basis uniquely corresponds to a direction $\mathbf{y} = \left[\begin{smallmatrix} \mathbf{y}_P \\ \mathbf{y}_Q \end{smallmatrix}\right]$. Observe that subspace basis $\boldsymbol{\Lambda}_r$ assumes, just like traditional simplex algorithm basis $\mathbf{A}_B$, that all variables it contains are free to move in either direction. This assumption could be proven false since some residual upper bounds of the $\mathbf{y}_P$-variables *could* be zero (see Section 4 for details on the content of P). Therefore, the necessary and sufficient optimality conditions of Proposition 1 cannot be used in the generic algorithm.

We resort to the usual sufficient optimality conditions which state that current solution $\vec{\mathbf{y}}_P = \overleftarrow{\mathbf{y}}_P = \mathbf{0},\ \vec{\mathbf{y}}_{\vec{Q}} = \mathbf{0},\ \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} = \mathbf{0}$ to problem (9) is optimal or, equivalently, $\mathbf{x}^k$ is optimal for (1), if there exists some $\boldsymbol{\psi}^\intercal = [\boldsymbol{\psi}_R^\intercal, \boldsymbol{\psi}_S^\intercal]$ such that the reduced costs of the $\vec{\mathbf{y}}_P, \overleftarrow{\mathbf{y}}_P, \vec{\mathbf{y}}_{\vec{Q}}, \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}}$-variables are greater than or equal to zero. With respect to the $\vec{\mathbf{y}}_P, \overleftarrow{\mathbf{y}}_P$-variables, this assertion can only be validated if their reduced costs are both null. Indeed, on the one hand, these values must be nonnegative and on the other hand, they only differ by their signs. It turns out that the reduced cost of $\mathbf{y}_P$-variables is $\bar{\mathbf{c}}_P^\intercal = \mathbf{c}_P^\intercal - \boldsymbol{\psi}_R^\intercal \mathbf{A}_{RP} - \boldsymbol{\psi}_S^\intercal \bar{\mathbf{A}}_{SP}$. Given that $\bar{\mathbf{A}}_{SP} = \mathbf{0}$, a null value can only be achieved with $\boldsymbol{\psi}_R^\intercal := \mathbf{c}_P^\intercal \mathbf{A}_{RP}^{-1}$ regardless of $\boldsymbol{\psi}_S^\intercal$.

The pricing step takes the role of an oracle built in light of this consideration. The vector of dual variables $\boldsymbol{\psi}_R$ is fixed and it is thus possible to omit the nonnegativity checkup of the reduced costs of the $\mathbf{y}_P$-variables. The dual formulation seeks the smallest reduced cost $\mu^k \le 0$ of the $\vec{\mathbf{y}}_Q, \overleftarrow{\mathbf{y}}_Q$-variables by optimizing over the vector of dual variables $\boldsymbol{\psi}_S$:

$$
\begin{array}{rlll}
\max\limits_{\mu \le 0,\, \boldsymbol{\psi}_S} \mu \quad \text{s.t.} \quad \mu \le & (c_j - \mathbf{c}_P^\intercal \mathbf{A}_{RP}^{-1} \mathbf{A}_{Rj} - \boldsymbol{\psi}_S^\intercal \bar{\mathbf{A}}_{Sj}) & [\,\vec{y}_j\,] & \forall j \in \vec{Q} \\
\mu \le & -(c_j - \mathbf{c}_P^\intercal \mathbf{A}_{RP}^{-1} \mathbf{A}_{Rj} - \boldsymbol{\psi}_S^\intercal \bar{\mathbf{A}}_{Sj}) & [\,\overleftarrow{y}_j\,] & \forall j \in \overleftarrow{Q}.
\end{array}
\tag{10}
$$

Let $\tilde{\mathbf{c}}_Q^\intercal := \mathbf{c}_Q^\intercal - \boldsymbol{\psi}_R^\intercal \mathbf{A}_{RQ}$, be the partial reduced cost computed only using $\boldsymbol{\psi}_R^\intercal$. Taking the dual of (10) brings a primal formulation of the pricing problem which is written in terms of $\vec{\mathbf{y}}_{\vec{Q}}$ and $\overleftarrow{\mathbf{y}}_{\overleftarrow{Q}}$, the variables to price out and effectively searches for the Q-components of a weighted cycle of minimal reduced cost:

$$
\begin{array}{rlrcll}
\mu \; = \min & & \tilde{\mathbf{c}}_{\vec{Q}}^\intercal \vec{\mathbf{y}}_{\vec{Q}} & - & \tilde{\mathbf{c}}_{\overleftarrow{Q}}^\intercal \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} & \\
& \text{s.t.} & \mathbf{1}^\intercal \vec{\mathbf{y}}_{\vec{Q}} & + & \mathbf{1}^\intercal \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} & \le 1 \quad [\mu] \\
& & \bar{\mathbf{A}}_{S\vec{Q}} \vec{\mathbf{y}}_{\vec{Q}} & - & \bar{\mathbf{A}}_{S\overleftarrow{Q}} \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} & = \mathbf{0} \quad [\boldsymbol{\psi}_S] \\
& & \vec{\mathbf{y}}_{\vec{Q}} \ge \mathbf{0}, & & \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} \ge \mathbf{0}. &
\end{array}
\tag{11}
$$

Formulation (11) comprises $m-r+1$ constraints and is feasible as the slack variable in the inequality constraint can take value 1 at a null cost. Nonnull extreme point solutions to (11) are rays of the cone defined by $\{\vec{\mathbf{y}}_{\bar{Q}} \geq \mathbf{0},\ \overleftarrow{\mathbf{y}}_{\bar{Q}} \geq \mathbf{0} \mid \bar{\mathbf{A}}_{S\bar{Q}}\vec{\mathbf{y}}_{\bar{Q}} - \bar{\mathbf{A}}_{S\bar{Q}}\overleftarrow{\mathbf{y}}_{\bar{Q}} = \mathbf{0}\}$, the values of $\vec{\mathbf{y}}_{\bar{Q}}$ and $\overleftarrow{\mathbf{y}}_{\bar{Q}}$ being scaled by the normalizing convexity constraint. Let μ^k, $\vec{\mathbf{y}}^k_{\bar{Q}}$ and $\overleftarrow{\mathbf{y}}^k_{\bar{Q}}$ denote an optimal solution to the primal-dual system (10)–(11) at iteration k.

3.4. Direction $\mathbf{y}$

When $\mu^k = 0$, solution $\mathbf{x}^k$ is optimal for LP (1) and the algorithm terminates. Assume $\mu^k < 0$ and a partial direction $\mathbf{y}^k_Q = \begin{bmatrix} \vec{\mathbf{y}}^k_{Q_F} - \overleftarrow{\mathbf{y}}^k_{Q_F} \\ \vec{\mathbf{y}}^k_{Q_L} \\ -\overleftarrow{\mathbf{y}}^k_{Q_U} \end{bmatrix} \in \mathbb{R}^{n-r}$ is retrieved. The complete direction $\mathbf{y}^k = \begin{bmatrix} \mathbf{y}^k_P \\ \mathbf{y}^k_Q \end{bmatrix}$ is established in Step 4 of the algorithm with the orthogonal remainder $\mathbf{y}^k_P \in \mathbb{R}^r$. This is done by solving, within (9), the system

$$\begin{aligned} \mathbf{A}_{RP}(\vec{\mathbf{y}}_P - \overleftarrow{\mathbf{y}}_P) + (\mathbf{A}_{R\vec{Q}}\vec{\mathbf{y}}^k_{\vec{Q}} - \mathbf{A}_{R\bar{Q}}\overleftarrow{\mathbf{y}}^k_{\bar{Q}}) = \mathbf{0} \\ \vec{\mathbf{y}}_P \geq \mathbf{0},\ \overleftarrow{\mathbf{y}}_P \geq \mathbf{0}, \qquad \vec{y}_j\,\overleftarrow{y}_j = 0, \quad \forall j \in P. \end{aligned} \tag{12}$$

The orthogonal remainder $\mathbf{y}^k_P = -\bar{\mathbf{a}}^k_R := \mathbf{A}^{-1}_{RP}(\mathbf{A}_{R\vec{Q}}\vec{\mathbf{y}}^k_{\vec{Q}} - \mathbf{A}_{R\bar{Q}}\overleftarrow{\mathbf{y}}^k_{\bar{Q}})$ is unique. However, among the infinitely many possibilities for $(\vec{\mathbf{y}}^k_P - \overleftarrow{\mathbf{y}}^k_P) = \mathbf{y}^k_P$, only one remains by taking the positive and negative part of vector $-\bar{\mathbf{a}}^k_R$ for the respective $\vec{\mathbf{y}}^k_P, \overleftarrow{\mathbf{y}}^k_P$-variables:

$$\begin{aligned} \overleftarrow{\mathbf{y}}^k_{j_i} = \quad & \bar{a}^k_i, \text{ if } \bar{a}^k_i > 0, \quad \text{otherwise } 0, \quad i \in R,\ j_i \in P \\ \vec{\mathbf{y}}^k_{j_i} = \quad & -\bar{a}^k_i, \text{ if } \bar{a}^k_i < 0, \quad \text{otherwise } 0, \quad i \in R,\ j_i \in P. \end{aligned} \tag{13}$$

In summary, finding direction $\mathbf{y}^k \in \mathbb{R}^n$ with negative reduced cost μ^k is split in two parts. The first optimizes primal pricing problem (11) with the scaling convexity constraint for which extreme point solution $(\vec{\mathbf{y}}^k_{\bar{Q}}, \overleftarrow{\mathbf{y}}^k_{\bar{Q}})$ involves $m-r+1$ basic variables of the linear system in (11), some being degenerate at zero. Given partial direction $\mathbf{y}^k_Q \in \mathbb{R}^{n-r}$, the second part computes the changes on at most r variables from $\mathbf{y}^k_P \in \mathbb{R}^r$ in a space orthogonal to the first partial direction. In total, at most $m+1$ components are different from zero in direction vector $\mathbf{y}^k \in \mathbb{R}^n$:

$$\mathbf{y}^k = \begin{bmatrix} \mathbf{y}^k_P \\ \mathbf{y}^k_Q \end{bmatrix} = \begin{bmatrix} -\bar{\mathbf{a}}^k_R \\ \vec{\mathbf{y}}^k_{Q_F} - \overleftarrow{\mathbf{y}}^k_{Q_F} \\ \vec{\mathbf{y}}^k_{Q_L} \\ -\overleftarrow{\mathbf{y}}^k_{Q_U} \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{y}}^k_{P_F} - \overleftarrow{\mathbf{y}}^k_{P_F} \\ \vec{\mathbf{y}}^k_{P_L} - \overleftarrow{\mathbf{y}}^k_{P_L} \\ \vec{\mathbf{y}}^k_{P_U} - \overleftarrow{\mathbf{y}}^k_{P_U} \\ \vec{\mathbf{y}}^k_{Q_F} - \overleftarrow{\mathbf{y}}^k_{Q_F} \\ \vec{\mathbf{y}}^k_{Q_L} \\ -\overleftarrow{\mathbf{y}}^k_{Q_U} \end{bmatrix}. \tag{14}$$

The complementarity condition $\vec{\mathbf{y}}^\intercal\overleftarrow{\mathbf{y}} = 0$ is taken into account during both of these parts. First off, by looking for extreme point solutions to the pricing problem, any nonnull reduced cost weighted cycle cannot contain both directed $\mathbf{y}^k_Q$-variables simultaneously. Secondly, $\mathbf{y}^k_P$

is established in (13) with the only solution that forces at least one directed $\mathbf{y}_P$-variable at zero.

Implicit bounds. Solving (11) for obtaining the first partial direction can be interpreted as a Dantzig-Wolfe decomposition performed on the residual problem $LP(\mathbf{x}^k)$ (9). This has already been observed by Metrane et al. (2010) for the Improved Primal Simplex algorithm. In the proposed framework, the domain of the pricing step is given by $\{\vec{\mathbf{y}}_{\vec{Q}} \geq \mathbf{0},\ \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} \geq \mathbf{0} \mid \bar{\mathbf{A}}_{S\vec{Q}}\vec{\mathbf{y}}_{\vec{Q}} - \bar{\mathbf{A}}_{S\overleftarrow{Q}}\overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} = \mathbf{0}\}$ and a selected negative reduced cost extreme ray is normalized by the convexity constraint in (11). This means that if a variable $x_j,\ j \in P$, has an implicit lower bound $\underline{l}_j > l_j$ or an implicit upper bound $\bar{u}_j < u_j$ due to the technological constraints of LP, such a restriction is necessarily transferred to the reformulation. This is to say that when $x_j,\ j \in P$, is free to move forward and backward with respect to the explicit bounds $\ell_j \leq x_j \leq u_j$, the implicit bounds are properly handled through the equivalent reformulation thus meaning that no direction can go against them $\ell_j < \underline{\ell}_j \leq x_j \leq \bar{u}_j < u_j$ (Gauthier et al. 2015).

3.5. Exchange mechanism

Taking into account the column partition $\{P, Q\}$ linked with that of $\{F, L, U\}$, at most 10 nonzero residual upper bound vectors are involved in the computation of the step size. Hence, step size ρ^k in Step 5 of the generic algorithm is the solution of

$$\begin{aligned} \max\ \rho \quad \text{s.t. } \rho \begin{bmatrix} \vec{\mathbf{y}}^k_{P_F} \\ \vec{\mathbf{y}}^k_{P_L} \\ \vec{\mathbf{y}}^k_{P_U} \end{bmatrix} \leq \begin{bmatrix} \vec{\mathbf{r}}^k_{P_F} \\ \vec{\mathbf{r}}^k_{P_L} \\ \mathbf{0} \end{bmatrix}, \ \rho \begin{bmatrix} \overleftarrow{\mathbf{y}}^k_{P_F} \\ \overleftarrow{\mathbf{y}}^k_{P_L} \\ \overleftarrow{\mathbf{y}}^k_{P_U} \end{bmatrix} \leq \begin{bmatrix} \overleftarrow{\mathbf{r}}^k_{P_F} \\ \mathbf{0} \\ \overleftarrow{\mathbf{r}}^k_{P_U} \end{bmatrix} \\ \rho \begin{bmatrix} \vec{\mathbf{y}}^k_{Q_F} \\ \vec{\mathbf{y}}^k_{Q_L} \end{bmatrix} \leq \begin{bmatrix} \vec{\mathbf{r}}^k_{Q_F} \\ \vec{\mathbf{r}}^k_{Q_L} \end{bmatrix}, \ \rho \begin{bmatrix} \overleftarrow{\mathbf{y}}^k_{Q_F} \\ \overleftarrow{\mathbf{y}}^k_{Q_U} \end{bmatrix} \leq \begin{bmatrix} \overleftarrow{\mathbf{r}}^k_{Q_F} \\ \overleftarrow{\mathbf{r}}^k_{Q_U} \end{bmatrix}. \end{aligned} \tag{15}$$

A new primal solution is obtained by computing $\mathbf{x}' = \mathbf{x}^k + \rho^k \mathbf{y}^k$. Depending on the choice of the subspace basis $\mathbf{\Lambda}_r$, $\mathbf{x}'$ represented by $(\mathbf{x}'_F; \mathbf{x}'_L; \mathbf{x}'_U)$ could be nonbasic. Section 4.3 explains how, why and when this can happen with the conceptualization of interior directions. We simply mention that any nonbasic solution $\mathbf{x}'$ can be rendered basic by solving a restricted problem over the set of free variables:

$$\begin{aligned} z = \min \quad & \mathbf{c}_F^\intercal \mathbf{x}_F + \mathbf{c}_L^\intercal \mathbf{x}_L + \mathbf{c}_U^\intercal \mathbf{x}_U \\ \text{s.t.} \quad & \mathbf{A}_F \mathbf{x}_F + \mathbf{A}_L \mathbf{x}_L + \mathbf{A}_U \mathbf{x}_U = \mathbf{b} \\ & \mathbf{l}'_F \leq \mathbf{x}_F \leq \mathbf{u}'_F \\ & \mathbf{x}_L = \mathbf{x}'_L \\ & \mathbf{x}_U = \mathbf{x}'_U. \end{aligned} \tag{16}$$

This problem identifies weighted cycles comprising free variables only and increases or decreases the value of these variables until some lower and upper bounds are reached while

possibly improving the overall solution cost. In network terminology, one obtains a *cycle free solution*, that is, a network solution containing no cycle composed of free arcs only, see Ahuja et al. (1993).

Let $\mathbf{x}^{k+1}$ be the basic solution obtained from $\mathbf{x}'$ directly or after solving (16). Solution $\mathbf{x}^{k+1}$ can be partitioned as $[\mathbf{x}_F^{k+1}; \mathbf{x}_L^{k+1}; \mathbf{x}_U^{k+1}]$ where sets L and U at least contain the same variables as for $\mathbf{x}'$ as well as some elements of $\mathbf{x}'_F$ that are distributed across depending on the new solution. Remaining free variables of course belong to new set F.

Remark. The difference between problem (7) and problem (16) is subtle. The former establishes an independent row set $\mathbf{\Lambda}_R$ of $\mathbf{\Lambda}_r$. The latter ensures a basic solution and thus that the resulting columns in F necessarily verifies the independence assumption. Since $\mathbf{\Lambda}_r$ is typically the fruit of a mapping of $\mathbf{A}_F$, it could also be constructed on-the-fly with a nonbasic solution and additional independence verifications.

3.6. Data structures update

The final Step 6 updates all the appropriate data structures, namely current basic solution $\mathbf{x}^{k+1}$, objective function $z^{k+1} = \mathbf{c}^\intercal \mathbf{x}^{k+1}$, residual upper bounds $\vec{\mathbf{r}}^{k+1} = \mathbf{u} - \mathbf{x}^{k+1}$ and $\overleftarrow{\mathbf{r}}^{k+1} = \mathbf{x}^{k+1} - \mathbf{l}$ for $LP(\mathbf{x}^{k+1})$, column partitions $\{B, N\}$ and $\{F, L, U\}$. Another iteration $k \to k+1$ then starts in Step 1 by selecting a new subspace basis $\mathbf{\Lambda}_r$, $0 \le r \le m$.

4. Properties

In this section, we look at properties that are shared with any variant of the algorithm. In Section 4.1, we dive in particular environments of the generic algorithm and make connections with different known algorithms from the literature. The requirements to ensure nondegenerate pivots are discussed in Section 4.2 from which emanates a family of nondegenerate variants. While all members of this family bring necessary and sufficient optimality conditions to the table, Section 4.3 explains the reasons behind this perplexity with the help of interior directions. Finally, an illustrative example of three particular variants is presented in Section 4.4.

Recall that $\emptyset \subseteq P \subseteq B$ is the set of *basic* variables that are *compatible with* $\mathbf{\Lambda}_r$, that is, those that belong to $\mathbf{V}(\mathbf{\Lambda}_r)$. The analysis mainly focuses on convergent cases of the generic algorithm and properties are presented in terms of the content of set P. We start our analysis with a result establishing a fundamental difference between the classical pricing problem of finding a variable with the most negative reduced cost as in PS and that of optimizing the dual variable vector for determining the smallest possible reduced cost.

Proposition 3 *Let $\mathbf{x}^k$, $k \ge 0$, be a nonoptimal solution to LP (1) and $(\vec{\mathbf{y}}_{\vec{Q}}^k, \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}}^k)$ be an optimal solution to the primal pricing problem (11). There exists some solution $\boldsymbol{\psi}^\intercal = [\boldsymbol{\psi}_R^\intercal, \boldsymbol{\psi}_S^\intercal]$ to the dual pricing problem (10) such that the reduced cost of every positive variable in $(\vec{\mathbf{y}}_{\vec{Q}}^k, \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}}^k)$ is equal to the minimum reduced cost μ^k, that is,*

$$\begin{aligned} \vec{y}_j^k > 0, \quad j \in \vec{Q} \quad &\Rightarrow \quad \bar{c}_j = \mu^k \\ \overleftarrow{y}_j^k > 0, \quad j \in \overleftarrow{Q} \quad &\Rightarrow \quad -\bar{c}_j = \mu^k. \end{aligned} \tag{17}$$

Proof. By complementary slackness, at the optimality of the primal-dual pair (10)–(11) of the pricing problem, the equality holds to μ^k in (10) for every positive variable in the solution of (11). Hence, an associated optimal $\boldsymbol{\psi}_S^\intercal$ together with $\boldsymbol{\psi}_R^\intercal = \mathbf{c}_P^\intercal \mathbf{A}_{RP}^{-1}$ satisfy the statement of the proposition. □

4.1. Special cases

We present five non exhaustive special variants of the generic algorithm for which set P equals specific subsets of indices: B (all basic variables), F (only the free variables), $\emptyset$ (none of the basic variables), $F \subseteq P$ (at least the set of free variables), and finally $P \subseteq F$ (at most the set of free variables).

Table 1, which is divided in four groups, summarizes characteristics and properties for the variants analyzed. Further observe that among these, two remain susceptible to degenerate pivots such that their pricing problem is reminiscent of the residual problem $LP(\mathbf{x}^k)$ whereas the other three fall into the nondegenerate pivot family with a pricing problem that reflects the residual problem $RP(\mathbf{x}^k)$. The generic version is presented on the left alongside the components described.

In the first group, we find $\emptyset \subseteq P \subseteq B$ from which we derive dimension $0 \leq r \leq m$ of subspace basis $\boldsymbol{\Lambda}_r$. Note that r is the dimension of the *a posteriori* computed partial direction $\mathbf{y}_P \in \mathbb{R}^r$. In most variants, the definition of $\boldsymbol{\Lambda}_r = \begin{bmatrix} \boldsymbol{\Lambda}_R \\ \boldsymbol{\Lambda}_S \end{bmatrix}$ and the induced set P are interchangeable. As the subspace basis can also be written as $\begin{bmatrix} \mathbf{I}_r \\ \mathbf{M} \end{bmatrix}$, we give an expression for matrix $\mathbf{M}$ from which matrix $\mathbf{T}_r^{-1} = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ -\mathbf{M} & \mathbf{I}_{m-r} \end{bmatrix}$ can be derived. Comes last in that first group, the vector of fixed dual variables $\boldsymbol{\psi}_R^\intercal = \mathbf{c}_P^\intercal \mathbf{A}_{RP}^{-1}$ when optimizing for the minimum reduced cost μ.

The second group concerns the primal pricing problem (11) from which we compute the partial direction $\mathbf{y}_Q \in \mathbb{R}^{n-r}$, where set $Q = Q_F \cup Q_L \cup Q_U$. Its row size is $s+1 = m-r+1$ whereas the number of variables is given by $|\vec{Q} \cup \overleftarrow{Q}| = 2|Q_F| + |Q_L| + |Q_U|$.

In the third group, we examine the behavior of the objective function z which is either nonincreasing when degenerate pivots may occur or strictly decreasing otherwise. There is only one case for which the minimum reduced cost μ is nondecreasing from one iteration to the next ($P = \emptyset$), otherwise we observe oscillations of the successive values (see Section 4.4).

The final group provides the details of the column partition $\{P, Q\}$ of direction $\mathbf{y} = \begin{bmatrix} \mathbf{y}_P \\ \mathbf{y}_Q \end{bmatrix}$ for which iteration subscript k is discarded for conciseness.

Case $P = B$. Choosing m linearly independent column vectors, then $\boldsymbol{\Lambda}_m = \mathbf{I}_m \equiv \mathbf{A}_B$ and $\mathbf{V}(\boldsymbol{\Lambda}_m) = \mathbb{R}^m$, all basic variables are compatible and $Q = N$. It induces $\mathbf{T}_m = \mathbf{T}_m^{-1} = \begin{bmatrix} \mathbf{I}_m & \emptyset \\ \emptyset & \mathbf{I}_0 \end{bmatrix} = \mathbf{I}_m$ and, most importantly, fixes $\boldsymbol{\pi}^\intercal = \mathbf{c}_B^\intercal \mathbf{A}_B^{-1}$. The generic algorithm becomes PS with Dantzig's pivot-selection rule. For simplicity of the presentation, let us consider

Generic algorithm	**PS**	**DCA**	**IPS**	**GDL**	**MMCC**
	Variants with possible degenerate pivots		*Variants with nondegenerate pivots*		
	Residual problem: $LP(\mathbf{x}^k)$		*Residual problem:* $RP(\mathbf{x}^k)$		
P	$\emptyset \subseteq P \subseteq B$	B	$F \subseteq P$	F	$P \subseteq F$	$\emptyset$
$\lvert P\rvert = r$	$[0,m]$	m	$[f,m]$	f	$[0,f]$	0
$\mathbf{\Lambda}_r$	$\begin{bmatrix}\mathbf{\Lambda}_R\\ \mathbf{\Lambda}_S\end{bmatrix}$	$\mathbf{A}_B \equiv \mathbf{I}_m$	$\begin{bmatrix}\mathbf{I}_r\\ \mathbf{\Lambda}_S\end{bmatrix}$	$\mathbf{A}_F = \begin{bmatrix}\mathbf{A}_{RF}\\ \mathbf{A}_{SF}\end{bmatrix}$	$\mathbf{A}_{P_F} = \begin{bmatrix}\mathbf{A}_{RP_F}\\ \mathbf{A}_{SP_F}\end{bmatrix}$	$\mathbf{\Lambda}_0 = \emptyset$
$\mathbf{M}$	$\mathbf{\Lambda}_S\mathbf{\Lambda}_R^{-1}$	$\emptyset$	$\mathbf{\Lambda}_S$	$\mathbf{A}_{SF}\mathbf{A}_{RF}^{-1}$	$\mathbf{A}_{SP_F}\mathbf{A}_{RP_F}^{-1}$	$\emptyset$
$\boldsymbol{\psi}_R^\intercal$	$\mathbf{c}_P^\intercal\mathbf{A}_{RP}^{-1}$	$\mathbf{c}_B^\intercal\mathbf{A}_B^{-1}$	$\mathbf{c}_P^\intercal$	$\mathbf{c}_F^\intercal\mathbf{A}_{RF}^{-1}$	$\mathbf{c}_{P_F}^\intercal\mathbf{A}_{RP_F}^{-1}$	$\emptyset$
Q	$Q_F \cup Q_L \cup Q_U$	N	$Q_L \cup Q_U$	$L \cup U$	$Q_F \cup L \cup U$	$F \cup L \cup U$
$\lvert Q\rvert$	$n-r$	$n-m$	$[n-m, n-f]$	$n-f$	$[n-f, n]$	n
$s+1$	$m-r+1$	1	$[1, m-f+1]$	$m-f+1$	$[m-f+1, m+1]$	$m+1$
$\lvert\vec{Q}\cup\overleftarrow{Q}\rvert$	$2\lvert Q_F\rvert + \lvert Q_L\rvert + \lvert Q_U\rvert$	$n-m$	$[n-m, n-f]$	$n-f$	$[n-f, n+f]$	$n+f$
z	$z^k \geq z^{k+1}$	$z^k \geq z^{k+1}$	$z^k \geq z^{k+1}$	$z^k > z^{k+1}$	$z^k > z^{k+1}$	$z^k > z^{k+1}$
μ	*oscillations*	*oscillations*	*oscillations*	*oscillations*	*oscillations*	$\mu^k \leq \mu^{k+1}$
$\begin{bmatrix}\mathbf{y}_P\\ \mathbf{y}_Q\end{bmatrix}$	$\begin{bmatrix}\vec{\mathbf{y}}_{P_F} - \overleftarrow{\mathbf{y}}_{P_F}\\ \vec{\mathbf{y}}_{P_L} - \overleftarrow{\mathbf{y}}_{P_L}\\ \vec{\mathbf{y}}_{P_U} - \overleftarrow{\mathbf{y}}_{P_U}\\ \vec{\mathbf{y}}_{Q_F} - \overleftarrow{\mathbf{y}}_{Q_F}\\ \vec{\mathbf{y}}_{Q_L}\\ -\overleftarrow{\mathbf{y}}_{Q_U}\end{bmatrix}$	$\begin{bmatrix}\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F\\ \vec{\mathbf{y}}_{B_L} - \overleftarrow{\mathbf{y}}_{B_L}\\ \vec{\mathbf{y}}_{B_U} - \overleftarrow{\mathbf{y}}_{B_U}\\ \vec{\mathbf{y}}_{N_L}\\ -\overleftarrow{\mathbf{y}}_{N_U}\end{bmatrix}$	$\begin{bmatrix}\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F\\ \vec{\mathbf{y}}_{P_L} - \overleftarrow{\mathbf{y}}_{P_L}\\ \vec{\mathbf{y}}_{P_U} - \overleftarrow{\mathbf{y}}_{P_U}\\ \vec{\mathbf{y}}_{Q_L}\\ -\overleftarrow{\mathbf{y}}_{Q_U}\end{bmatrix}$	$\begin{bmatrix}\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F\\ \vec{\mathbf{y}}_L\\ -\overleftarrow{\mathbf{y}}_U\end{bmatrix}$	$\begin{bmatrix}\vec{\mathbf{y}}_{P_F} - \overleftarrow{\mathbf{y}}_{P_F}\\ \vec{\mathbf{y}}_{Q_F} - \overleftarrow{\mathbf{y}}_{Q_F}\\ \vec{\mathbf{y}}_L\\ -\overleftarrow{\mathbf{y}}_U\end{bmatrix}$	$\begin{bmatrix}\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F\\ \vec{\mathbf{y}}_L\\ -\overleftarrow{\mathbf{y}}_U\end{bmatrix}$

Table 1 Characteristics and properties for different variations of the generic algorithm

a linear program in standard form ($U = \emptyset$), that is, $z^\star := \min\{\mathbf{c}^\intercal\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$, for which the residual problem $LP(\mathbf{x}^k)$ takes the simpler form

$$
\begin{aligned}
z^\star := \mathbf{c}^\intercal\mathbf{x}^k \ + \quad \min \quad & \mathbf{c}_B^\intercal(\vec{\mathbf{y}}_B - \overleftarrow{\mathbf{y}}_B) + \mathbf{c}_N^\intercal\vec{\mathbf{y}}_N \\
\text{s.t.} \quad & \mathbf{A}_B(\vec{\mathbf{y}}_B - \overleftarrow{\mathbf{y}}_B) + \mathbf{A}_N\vec{\mathbf{y}}_N \ = \mathbf{0} \quad [\boldsymbol{\pi}] \\
& \vec{\mathbf{y}}_B \geq \mathbf{0},\ \overleftarrow{\mathbf{y}}_B \geq \mathbf{0} \\
& \overleftarrow{\mathbf{y}}_F \leq \mathbf{x}_F^k,\ \overleftarrow{\mathbf{y}}_{B_L} \leq \mathbf{0} \\
& \vec{\mathbf{y}}_N \geq \mathbf{0}.
\end{aligned}
\tag{18}
$$

Here, $P = F \cup B_L$ and there is no upper bounds on the nonbasic variables $\vec{\mathbf{y}}_N \geq \mathbf{0}$ whereas two components of $\overleftarrow{\mathbf{y}}_B \geq \mathbf{0}$ are upper bounded, namely $\overleftarrow{\mathbf{y}}_F \leq \mathbf{x}_F^k$ and $\overleftarrow{\mathbf{y}}_{B_L} \leq \mathbf{0}$. We are looking for a potentially improving direction $\mathbf{y} = \begin{bmatrix}\mathbf{y}_B\\ \mathbf{y}_N\end{bmatrix} = \begin{bmatrix}\vec{\mathbf{y}}_B - \overleftarrow{\mathbf{y}}_B\\ \vec{\mathbf{y}}_N\end{bmatrix}$. One easily recognizes the poor contribution of the partial direction $\vec{\mathbf{y}}_N$ based only on the reduced cost values and the cone formed by $\vec{\mathbf{y}}_N \geq \mathbf{0}$, without considering any other constraints of the residual problem (18). Indeed, finding the smallest reduced cost over $\bar{\mathbf{c}}_N^\intercal = \mathbf{c}_N^\intercal - \boldsymbol{\pi}^\intercal\mathbf{A}_N$ can be

formulated as

$$\begin{array}{lll} \max & \mu & \\ \text{s.t.} & \mu \le \bar{c}_j & [\vec{y}_j] \quad \forall j \in N \\ & \mu \ge 0. & \end{array} \tag{19}$$

Taking the dual, one selects a single entering variable with the smallest negative reduced cost, if any, as an extreme point solution of

$$\begin{array}{lll} \mu = \min & \displaystyle\sum_{j \in N} \bar{c}_j \vec{y}_j & \\ \text{s.t.} & \displaystyle\sum_{j \in N} \vec{y}_j \le 1 & [\mu] \\ & \vec{y}_j \ge 0 \qquad \forall j \in N. & \end{array} \tag{20}$$

The row size of the primal pricing problem is $s+1=1$ as it disregards the set of equality constraints in (18). These are only taken into account *a posteriori* when evaluating the impact of $\vec{\mathbf{y}}_N$ on the basic variables $(\vec{\mathbf{y}}_B - \overleftarrow{\mathbf{y}}_B)$. Indeed, when $\mu < 0$, vector $\vec{\mathbf{y}}_N$ contains a single variable at 1, say $\vec{y}_\ell$, $\ell \in N$ (the entering variable). Its impact on the basic variables is obtained by solving $(\vec{\mathbf{y}}_B - \overleftarrow{\mathbf{y}}_B) + \bar{\mathbf{a}}_\ell = \mathbf{0}$, $\vec{\mathbf{y}}_B \ge \mathbf{0}$, $\overleftarrow{\mathbf{y}}_B \ge \mathbf{0}$, $\vec{\mathbf{y}}_B^\intercal \overleftarrow{\mathbf{y}}_B = 0$. This is followed by the computation of the step size where vector $\overleftarrow{\mathbf{y}}_{B_L} \le \mathbf{0}$ may force a degenerate pivot. Therefore, the objective function is nondecreasing and PS may even not converge. Finally, we observe an oscillating behavior for μ, see the illustrative example of Section 4.4.

Case $P = F$. Choosing $P = F$ corresponds to selecting $\mathbf{\Lambda}_f \equiv \mathbf{A}_F$ and is the strategy developed by Elhallaoui et al. (2011) in the IPS algorithm. This induces $Q = L \cup U$, that is, the vector space $\mathbf{V}(\mathbf{\Lambda}_f)$ of compatible vectors does include all columns of $\mathbf{A}_F = \left[\begin{smallmatrix} \mathbf{A}_{RF} \\ \mathbf{A}_{SF} \end{smallmatrix}\right]$ but none associated with the degenerate basic variables: $\forall j \in B,\ \mathbf{a}_j \in \mathbf{V}(\mathbf{\Lambda}_f) \Leftrightarrow j \in F$. The linear transformation is given by $\mathbf{T}_f^{-1} = \left[\begin{smallmatrix} \mathbf{I}_f & \mathbf{0} \\ -\mathbf{M} & \mathbf{I}_{m-f} \end{smallmatrix}\right]$ where $\mathbf{M} = \mathbf{A}_{SF}\mathbf{A}_{RF}^{-1}$. Formulation $LP(\mathbf{x}^k)$ (9) is as follows:

$$\begin{array}{lllllll} z^\star := \mathbf{c}^\intercal \mathbf{x}^k & + & \min & \mathbf{c}_F^\intercal(\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F) & + & \mathbf{c}_L^\intercal \vec{\mathbf{y}}_L - \mathbf{c}_U^\intercal \overleftarrow{\mathbf{y}}_U & \\ & & \text{s.t.} & \mathbf{A}_{RF}(\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F) & + & \mathbf{A}_{RL}\vec{\mathbf{y}}_L - \mathbf{A}_{RU}\overleftarrow{\mathbf{y}}_U = \mathbf{0} & [\boldsymbol{\psi}_R] \\ & & & \vec{\mathbf{y}}_F \ge \mathbf{0},\ \overleftarrow{\mathbf{y}}_F \ge \mathbf{0} & & & \\ & & & \vec{\mathbf{y}}_F \le \vec{\mathbf{r}}_F^k,\ \overleftarrow{\mathbf{y}}_F \le \overleftarrow{\mathbf{r}}_F^k, & & \vec{\mathbf{y}}_L \le \vec{\mathbf{r}}_L^k,\ \overleftarrow{\mathbf{y}}_U \le \overleftarrow{\mathbf{r}}_U^k & \\ & & & & & \bar{\mathbf{A}}_{SL}\vec{\mathbf{y}}_L - \bar{\mathbf{A}}_{SU}\overleftarrow{\mathbf{y}}_U = \mathbf{0} & [\boldsymbol{\psi}_S] \\ & & & & & \vec{\mathbf{y}}_L \ge \mathbf{0},\ \overleftarrow{\mathbf{y}}_U \ge \mathbf{0}. & \end{array} \tag{21}$$

At this point, observe that all the residual upper bounds considered in the residual problem (21) are positive, thus it is explicitly equivalent to $RP(\mathbf{x}^k)$ (4). IPS fixes the dual variables corresponding to the free variables using the working basis $\mathbf{A}_{RF}$, that is, $\boldsymbol{\psi}_R^\intercal = \mathbf{c}_F^\intercal \mathbf{A}_{RF}^{-1}$, and optimizes $\boldsymbol{\psi}_S^\intercal$ to find a convex combination of the variables in $\vec{\mathbf{y}}_L$ and $\overleftarrow{\mathbf{y}}_U$

with the smallest reduced cost μ. The domain of the pricing problem $\{\vec{\mathbf{y}}_L \geq \mathbf{0},\ \overleftarrow{\mathbf{y}}_U \geq \mathbf{0} \mid \bar{\mathbf{A}}_{SL}\vec{\mathbf{y}}_L - \bar{\mathbf{A}}_{SU}\overleftarrow{\mathbf{y}}_U = \mathbf{0},\ \mathbf{1}^\intercal\vec{\mathbf{y}}_L + \mathbf{1}^\intercal\overleftarrow{\mathbf{y}}_U \leq 1\}$ adequately manages all *non free* variables indexed by $Q = L \cup U$ and the impact of the partial direction $\begin{bmatrix} \vec{\mathbf{y}}_L \\ -\overleftarrow{\mathbf{y}}_U \end{bmatrix}$ on those indexed by F, that is, $(\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F)$, is only on free variables that can move both forward and backward. Direction $\mathbf{y} = \begin{bmatrix} \vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F \\ \vec{\mathbf{y}}_L \\ -\overleftarrow{\mathbf{y}}_U \end{bmatrix}$ therefore induces a positive step size ρ^k because $\vec{\mathbf{r}}_F^k$, $\overleftarrow{\mathbf{r}}_F^k$, $\vec{\mathbf{r}}_L^k$, and $\overleftarrow{\mathbf{r}}_U^k$ are positive residual upper bound vectors. Objective function is strictly decreasing at every iteration and the Improved Primal Simplex algorithm converges. Finally, we still observe oscillations for μ, see again the illustrative example of Section 4.4.

Case $P = \emptyset$. Compared to $\mathbf{V}(\mathbf{\Lambda}_m) = \mathbb{R}^m$, this case $\mathbf{V}(\mathbf{\Lambda}_0) = \mathbf{0}$ reaches the other extreme: it only contains the null vector and there are no compatible variables. Complementary set $Q = F \cup L \cup U$ and, as in the case $P = B$, matrix $\mathbf{T}_0 = \mathbf{T}_0^{-1} = \begin{bmatrix} \mathbf{I}_0 & \emptyset \\ \emptyset & \mathbf{I}_m \end{bmatrix} = \mathbf{I}_m$. However, none of the dual variables is fixed ($r = 0$) and the m-dimensional original vector $\boldsymbol{\pi}$ is fully optimized to find the best negative reduced cost μ. The residual problem formulation $RP(\mathbf{x}^k)$ is the one already presented in (4) and the pricing problem comprises all the equality constraints, but no residual upper bounds (which are all positive). The domain of the pricing problem is therefore the cone formed by

$$\{\vec{\mathbf{y}}_F, \overleftarrow{\mathbf{y}}_F \geq \mathbf{0},\ \vec{\mathbf{y}}_L \geq \mathbf{0},\ \overleftarrow{\mathbf{y}}_U \geq \mathbf{0} \mid \mathbf{A}_F(\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F) + \mathbf{A}_L\vec{\mathbf{y}}_L - \mathbf{A}_U\overleftarrow{\mathbf{y}}_U = \mathbf{0}\} \tag{22}$$

and the convexity constraint. Direction $\mathbf{y} = \begin{bmatrix} \vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F \\ \vec{\mathbf{y}}_L \\ -\overleftarrow{\mathbf{y}}_U \end{bmatrix}$ is found at once by the pricing problem while the step size is determined using the upper bounds.

This particular case is one of the members of the family susceptible to lead to nonbasic solutions (see Section 4.3). However, since $\mathbf{T}_0^{-1} = \mathbf{I}_m$ and all dual variables are optimized in the pricing problem, the simplifications applicable to the different steps of the generic algorithm in Figure 2 imply that maintaining the basic nature of solutions is irrelevant. This strategy corresponds to the remarkable, strongly polynomial, Minimum Mean Cycle-Canceling algorithm for the capacitated network flow problems (Goldberg and Tarjan 1989, Radzik and Goldberg 1994). As for the case $\mathbf{\Lambda}_f \equiv \mathbf{A}_F$ (where the *a posteriori* affected variables in $P = F$ are only the free ones), the objective function strictly decreases at every iteration. Regarding μ, it is nondecreasing ($\mu^k \leq \mu^{k+1}$) with a reliable jump towards zero at the end of every so-called phase of at most $|A|$ iterations, where $|A|$ is the number of arcs in the network. See Gauthier et al. (2014b) for a complexity analysis of MMCC based on linear programming arguments and an improved complexity result over that of Radzik and Goldberg (1994). As shown in the next proposition, this nondecreasing property of μ is also true for linear programs.

Proposition 4 *Let $\mathbf{x}^k, k \geq 0$, be a nonoptimal solution. Given $P = \emptyset$, then $\mu^{k+1} \geq \mu^k$.*

Proof. Given $P = \emptyset$, let $(\mu^k, \boldsymbol{\pi}^k)$ be an optimal solution to the dual pricing problem (10) at iteration k, where $\mu^k < 0$ is the smallest reduced cost. By Proposition 3, μ^k is shared

by all the variables in the associated weighted cycle ω. The cost c_ω and the reduced cost $\bar{c}_\omega$ being the same, $\boldsymbol{\pi}^k$ can also be used while solving for μ^{k+1}. In the construction of $RP(\mathbf{x}^{k+1})$, some $\mathbf{y}$-variables are deleted and some are added compared to the set used in $RP(\mathbf{x}^k)$. Added variables can only be the reverse of those involved in ω, that is, some variables at their bounds become free or some variables reach one of their bounds. For this subset of added variables, the orientation is reversed and their reduced cost becomes $-\mu^k > 0$. Hence, all reduced costs are still greater than or equal to μ^k at iteration $k+1$ and μ^{k+1} cannot be less than the smallest coefficient in the objective function, that is, $\mu^{k+1} \geq \mu^k$. □

Case $F \subseteq P$. As $P \subseteq B$, this case is a compromise between algorithms PS ($P = B$) and IPS ($P = F$). The subspace basis $\boldsymbol{\Lambda}_r$, of possibly larger dimension $r \geq f$, is sufficient to span $\mathbf{A}_F$. At the risk of allowing degenerate pivots, this larger subspace basis gives a lot of freedom in the implementation of IPS and, more importantly, closes the theoretical gap between IPS and the Dynamic Constraint Aggregation method, see Gauthier et al. (2015). In fact, if the index set $P \cap \{L \cup U\} \neq \emptyset$ at a given iteration, then a degenerate pivot may occur. Again, μ shows no predictable behavior. Additional properties and characteristics are presented in Table 1.

Prior to the design of IPS, such a vector subspace basis was used for solving the linear relaxation of set partitioning problems (SPP) with DCA embedded in a column generation scheme (Elhallaoui et al. 2005, 2008). Let us consider the main aspects of this algorithm. Formulated as a binary linear program

$$\min \quad \mathbf{c}^\intercal \mathbf{x} \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{1}, \quad \mathbf{x} \in \mathbb{B}^n, \tag{23}$$

where $\mathbf{A} \in \mathbb{B}^{m \times n}$, SPP can be seen as a generic model encountered in various applications, namely, in vehicle routing and crew scheduling, see Desrosiers et al. (1995) and Desaulniers et al. (1998). DCA capitalizes on the compatibility interpretation and characterization of set partitioning optimal binary solutions. These are usually highly degenerate. Indeed, in typical applications, a vehicle route or a crew schedule covers several tasks, say on average $\bar{m}$, which implies that the number of variables assuming value one in the basis is of the order $m/\bar{m}$. The idea of row *aggregation* is born.

Columns are generated according to a given partition of the rows into *clusters*: this results in a row reduction of the master problem (23) since only a single row of each cluster is used. Note that a row clustering can be derived from any subset of at most m columns by identifying the identical rows. In DCA, aggregated columns are said to be *compatible with the row clustering*.

Assume that such a partition into r clusters is given and reorder the rows of the partition such that one row of each cluster appears in $\boldsymbol{\Lambda}_R = \mathbf{I}_r$. The next figure depicts an example with $r = 3$ clusters, where ($\mapsto$) consists of a reorganization of the lines such that the top rows in $\boldsymbol{\Lambda}_R$ are unique and the bottom rows in $\boldsymbol{\Lambda}_S = \mathbf{M}$ are duplicates, where every column of the binary matrix $\mathbf{M}$ comprises the remaining rows of a cluster:

$$\overset{Clusters}{\begin{pmatrix} 1 & & \\ 1 & & \\ 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & & 1 \\ & & 1 \\ & & 1 \end{pmatrix}} \mapsto \overset{\mathbf{\Lambda}_r}{\left(\begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \\ \hline 1 & & \\ 1 & & \\ 1 & & \\ & 1 & \\ & & 1 \\ & & 1 \end{array}\right)}.$$

DCA is a method steered by practical imperatives. The subspace basis $\mathbf{\Lambda}_r$ is obtained by design rather than as an equivalent form of the columns of $\mathbf{A}$ meaning that the intermediate operation goes unnoticed. In particular, observe that $\mathbf{M} = \mathbf{\Lambda}_S \mathbf{\Lambda}_R^{-1} = \mathbf{\Lambda}_S$. As $\mathbf{M}$ comprises the remaining rows of the clusters represented in $\mathbf{I}_r$, a column $\mathbf{a}_j$ of $\mathbf{A}$ is compatible with the row clustering if and only if $\bar{\mathbf{a}}_{Sj} := \mathbf{a}_{Sj} - \mathbf{M}\mathbf{a}_{Rj} = \mathbf{0}$. Compatibility thus means that the generated itineraries or schedules are those which precisely respect the current clustering into multi-task activities. Computational results also reveal that there is no inevitable correlation between an algorithm's inefficiency and degeneracy, contradicting common belief. See for example Benchimol et al. (2012) for the implementation of a stabilized DCA on highly degenerate multi-depot vehicle scheduling problems.

Case $P \subseteq F$. Denoted **GDL** in Table 1, this case contains all members for which the generic algorithm yields *a nondegenerate pivot at every iteration*. It includes IPS for which $P = F$ as well as the extreme case $P = \emptyset$. Although the objective function is strictly decreasing at every iteration, the minimum reduced cost μ can oscillate for $P \neq \emptyset$. Let us examine these variants in the following section.

4.2. Variants with nondegenerate pivots

Algorithms derived from the vector space decomposition framework can be divided in two families according to set P induced by the selected vector subspace basis $\mathbf{\Lambda}_r$. Algorithm family with $P \cap \{L \cup U\} \neq \emptyset$ may lead to degenerate pivots because two vectors are misleading the search for a strictly improving direction, that is, $\vec{\mathbf{y}}^k_{P_U} \leq \mathbf{0}$ and $\overleftarrow{\mathbf{y}}^k_{P_L} \leq \mathbf{0}$ for which the residual upper bounds are unfortunately only involved *a posteriori* in the step size computation. Family with $P \cap \{L \cup U\} = \emptyset$, or equivalently $\emptyset \subseteq P \subseteq F$, ensures a nondegenerate pivot at every iteration as shown in the next proposition.

Proposition 5 *Let $\mathbf{x}^k, k \geq 0$, be a nonoptimal basic solution to LP (1). If $\mathbf{\Lambda}_r$, $0 \leq r \leq f$, induces $P \cap \{L \cup U\} = \emptyset$ (or equivalently $P \subseteq F$), then the step size is strictly positive and a nondegenerate pivot occurs.*

Proof. Consider first the case $P = F$ where the variables indexed by L and U form set Q. That is, all variables at their lower or upper bounds without the distinction of being basic or nonbasic are evaluated first while computing the partial direction $\mathbf{y}_Q = \begin{bmatrix} \vec{\mathbf{y}}_L \\ -\overleftarrow{\mathbf{y}}_U \end{bmatrix}$. The

impact on vector $\mathbf{y}_P = (\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F)$ is computed second where only free variables are involved. Column partition $\{P, Q\} = \{F, L \cup U\}$ thus ensures a strict improvement of the objective function when $\mu < 0$. Consider now the case $P \subset F$: it induces partition $\{P_F, Q_F \cup L \cup U\}$ of the variables and also ensures nondegenerate pivots as the impact on vector $(\vec{\mathbf{y}}_{P_F} - \overleftarrow{\mathbf{y}}_{P_F})$ again only involves free variables. Hence, when $\mu < 0$ and $P \subseteq F$,

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_P \\ \mathbf{y}_Q \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{y}}_{P_F} - \overleftarrow{\mathbf{y}}_{P_F} \\ \vec{\mathbf{y}}_{Q_F} - \overleftarrow{\mathbf{y}}_{Q_F} \\ \vec{\mathbf{y}}_{Q_L} \\ -\overleftarrow{\mathbf{y}}_{Q_U} \end{bmatrix}, \tag{24}$$

and all positive variables of the selected direction $\mathbf{y}$ only involve positive residual upper bounds from which is computed a positive step size ρ, that is, $\rho\,\mu < 0$, and the objective function strictly decreases. □

The following proposition extends Proposition 1 and gives insight as to the reason for which both pricing problems that come from cases $P = F$ and $P = \emptyset$ provide necessary and sufficient optimality conditions.

Proposition 6 *Given $P \subseteq F$, a feasible solution $\mathbf{x}^k$ to* LP (1) *is optimal if and only if the following two equivalent conditions are satisfied:*

Primal: *pricing problem* (11) *contains no solution $(\vec{\mathbf{y}}^k_{\bar{Q}}, \overleftarrow{\mathbf{y}}^k_{\bar{Q}})$ of negative reduced cost.*

Dual: *$\exists \boldsymbol{\psi}_S$ solution to pricing problem* (10) *such that $\bar{\mathbf{c}}^\intercal_{\vec{Q}} \geq \mathbf{0}$, $\bar{\mathbf{c}}^\intercal_{\overleftarrow{Q}} \geq \mathbf{0}$ with $\boldsymbol{\psi}^\intercal_R = \mathbf{c}^\intercal_P \mathbf{A}^{-1}_{RP}$.*

Proof. We show that these two conditions are respectively equivalent to the primal and dual optimality conditions of Proposition 1. Given that $P \subseteq F$, variables $\vec{y}_j, \overleftarrow{y}_j, j \in P$, have positive residual upper bounds. Both are therefore in $RP(\mathbf{x}^k)$ and must necessarily have a null reduced cost to verify the nonnegativity requirement of Proposition 1. Because $\bar{\mathbf{A}}_{SP} = \mathbf{0}$, we have that $\bar{\mathbf{c}}^\intercal_P = \mathbf{c}^\intercal_P - \boldsymbol{\psi}_R \mathbf{A}_{RP} = \mathbf{0}$ and this can solely be achieved with $\boldsymbol{\psi}^\intercal_R = \mathbf{c}^\intercal_P \mathbf{A}^{-1}_{RP}$. Additionally requiring $\bar{\mathbf{c}}^\intercal_{\vec{Q}} \geq \mathbf{0}$ and $\bar{\mathbf{c}}^\intercal_{\overleftarrow{Q}} \geq \mathbf{0}$ satisfies the dual optimality condition of Proposition 1.

Regarding the primal condition, the objective function in (11) computes

$$\begin{aligned} \tilde{\mathbf{c}}^\intercal_{\vec{Q}} \vec{\mathbf{y}}_{\vec{Q}} - \tilde{\mathbf{c}}^\intercal_{\overleftarrow{Q}} \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} &= \bar{\mathbf{c}}^\intercal_{\vec{Q}} \vec{\mathbf{y}}_{\vec{Q}} - \bar{\mathbf{c}}^\intercal_{\overleftarrow{Q}} \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} \\ &= \bar{\mathbf{c}}^\intercal_P (\vec{\mathbf{y}}_P - \overleftarrow{\mathbf{y}}_P) + \bar{\mathbf{c}}^\intercal_{\vec{Q}} \vec{\mathbf{y}}_{\vec{Q}} - \bar{\mathbf{c}}^\intercal_{\overleftarrow{Q}} \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}} \\ &= \mathbf{c}^\intercal_P (\vec{\mathbf{y}}_P - \overleftarrow{\mathbf{y}}_P) + \mathbf{c}^\intercal_{\vec{Q}} \vec{\mathbf{y}}_{\vec{Q}} - \mathbf{c}^\intercal_{\overleftarrow{Q}} \overleftarrow{\mathbf{y}}_{\overleftarrow{Q}}, \end{aligned} \tag{25}$$

where the validity of the first equality follows from the homogeneous system in (11), adding the P-component with $\bar{\mathbf{c}}_P = \mathbf{0}$ gives us the second whereas the third equality reminds that the cost and the reduced cost of a weighted cycle is the same in $RP(\mathbf{x}^k)$. □

Remark. The argument in Section 3.3, while sufficient, is too stringent for the $\mathbf{y}_P$-variables since the variables in P that do not belong in F would be removed from the residual

problem $RP(\mathbf{x}^k)$ and their reduced costs are therefore irrelevant thus giving more freedom to $\boldsymbol{\psi}_R$.

While one might rest uneasy about the equivalent necessary and sufficient optimality conditions provided by two different oracles, the following section sheds light on their content.

4.3. Interior directions

Since this paper backtracks on the edge movement induced by a pivot by first considering the direction of travel, let us add a layer of definition on the resulting impact of this direction.

Definition 2 *Let C be a convex polyhedron. Given a vertex $\mathbf{x} \in C$ and vector $\mathbf{y} \neq \mathbf{0}$, let $\mathbf{x} + \rho\mathbf{y} \in C$ for some $\rho > 0$. Vector $\mathbf{y}$ is called an* edge direction *originating from $\mathbf{x}$ if for $0 < \delta < \rho$, vector $\mathbf{x} + \delta\mathbf{y}$ belongs to an edge of C. Otherwise, a nonedge direction is called an* interior direction *originating from $\mathbf{x}$.*

An important property of IPS is its movement on an edge of the polyhedron defined by the set of constraints of LP (1) at every iteration. The main idea of the proof is as follows (see Elhallaoui et al. 2011, Proposition 4). Consider a generic basis composed of the columns of $\mathbf{A}_F$ completed with $m - f$ artificial variables at zero. Either a single compatible variable or a combination of at most $m - f + 1$ incompatible variables at their lower or upper bounds is selected in the weighted cycle. In the former case, it acts as in PS with a nondegenerate pivot, hence a movement along an edge. In the latter case, the selected incompatible variables can enter the current basis one by one with degenerate pivots, each pivot removing an artificial variable thus maintaining the basis status of the solution, whereas the last pivot is nondegenerate, hence the movement is made along an edge. Proposition 5 shows that the family of algorithms with $P \cap \{L \cup U\} = \emptyset$, or equivalently $P \subseteq F$, ensures nondegenerate pivots at every iteration. One question that the reader may have in mind is *What differentiates Case $P = F$ from Case $P \subset F$?* The next proposition characterizes improving interior directions originating from a nonoptimal basic solution $\mathbf{x}^k$.

Proposition 7 *Let $\mathbf{x}^k$, $k \geq 0$, be a nonoptimal basic solution to LP (1). For $P \subset F$, the interior directions are nonnegative combinations of the edge directions.*

Proof. Residual problem $LP(\mathbf{x}^k)$ (4) can use the column index partition $\{F, L \cup U\}$ and hence be written as (21), as in IPS ($P = F$). Given an extreme point $(\vec{\mathbf{y}}_L \geq \mathbf{0}, \overleftarrow{\mathbf{y}}_U \geq \mathbf{0})$ solution to the pricing problem defined at $\mathbf{x}^k$, the remaining component is $(\vec{\mathbf{y}}_F - \overleftarrow{\mathbf{y}}_F) = -\mathbf{A}_{RF}^{-1}(\mathbf{A}_{RL}\vec{\mathbf{y}}_L - \mathbf{A}_{RU}\overleftarrow{\mathbf{y}}_U)$. Therefore, any strictly feasible direction $\mathbf{y}$ from vertex $\mathbf{x}^k$ takes the form $\mathbf{y} = \begin{bmatrix} -\mathbf{A}_{RF}^{-1}(\mathbf{A}_{RL}\vec{\mathbf{y}}_L - \mathbf{A}_{RU}\overleftarrow{\mathbf{y}}_U) \\ \vec{\mathbf{y}}_L \\ -\overleftarrow{\mathbf{y}}_U \end{bmatrix}$.

Using the Dantzig-Wolfe decomposition principle, formulation (21) can also be written in terms of the finite set Ω_F^k of extreme rays of $\{\vec{\mathbf{y}}_L \geq \mathbf{0},\ \overleftarrow{\mathbf{y}}_U \geq \mathbf{0} \mid \bar{\mathbf{A}}_{SL}\vec{\mathbf{y}}_L - \bar{\mathbf{A}}_{SU}\overleftarrow{\mathbf{y}}_U = \mathbf{0}\}$.

Therefore, any solution $\begin{bmatrix} \vec{\mathbf{y}}_L \\ \bar{\mathbf{y}}_U \end{bmatrix}$ to the pricing problem can be written as a nonnegative combination of these extreme rays

$$\begin{bmatrix} \vec{\mathbf{y}}_L \\ \bar{\mathbf{y}}_U \end{bmatrix} = \sum_{\omega \in \Omega_F^k} \begin{bmatrix} \vec{\mathbf{y}}_L^\omega \\ \bar{\mathbf{y}}_U^\omega \end{bmatrix} \rho^\omega, \qquad \forall \rho^\omega \geq 0,\ \forall \omega \in \Omega_F^k, \tag{26}$$

as well as any strictly feasible direction $\mathbf{y}$ from vertex $\mathbf{x}^k$ can be written as

$$\mathbf{y} = \sum_{\omega \in \Omega_F^k} \begin{bmatrix} -\mathbf{A}_{RF}^{-1}(\mathbf{A}_{RL}\vec{\mathbf{y}}_L^\omega - \mathbf{A}_{RU}\bar{\mathbf{y}}_U^\omega) \\ \vec{\mathbf{y}}_L^\omega \\ -\bar{\mathbf{y}}_U^\omega \end{bmatrix} \rho^\omega, \qquad \forall \rho^\omega \geq 0,\ \forall \omega \in \Omega_F^k. \tag{27}$$

For $P \subseteq F$, the extreme point solutions (weighted cycles) of the pricing problem are in a one-to-one correspondence with the extreme rays of the cone defined by removing the convexity constraint, hence in a one-to-one correspondence with strictly feasible directions (Proposition 5). Let Ω_P^k be the set of these extreme ray solutions. Therefore, for $P \subset F$, if $|\Omega_F^k| < |\Omega_P^k|$, every interior direction found in $\Omega_P^k \setminus \Omega_F^k$ is a nonnegative combination of the edge directions of Ω_F^k. □

While Proposition 7 is not fundamentally new, the partition scheme provided herein provides a way to identify interior directions. Their building block comes from the IPS construction which exhaustevily identifies the set of strictly feasible edge directions. Finally, we refer back to Section 3.5 where a direction leading to a nonbasic solution is evoked which can only happen with interior directions.

4.4. Illustrative example

This section presents a capacitated network flow problem on which PS, IPS and MMCC are applied. This allows to better understand the behavior of these algorithms and also to closely examine the structure of their pricing problems.

Data. Figure 3 shows on the left-hand side the original network $G = (V, E)$, where V is the set of nodes and E is the set of arcs. Four units are available at the central node 0 and one unit is required at all four other nodes. Cost values and flow bounds appear next to the arcs. We assume cost $c < 0$, hence the optimal solution on the right-hand side sends one unit of flow on the radial arcs $(0,1),(0,2),(0,3),(0,4)$ and 100 units on the counterclockwise ring arcs $(1,2),(2,3),(3,4),(4,1)$. Optimal cost is therefore $z^\star = 400c$ at vertex $\mathbf{x}^\star = (1,1,1,1,100,100,100,100)^\intercal$.

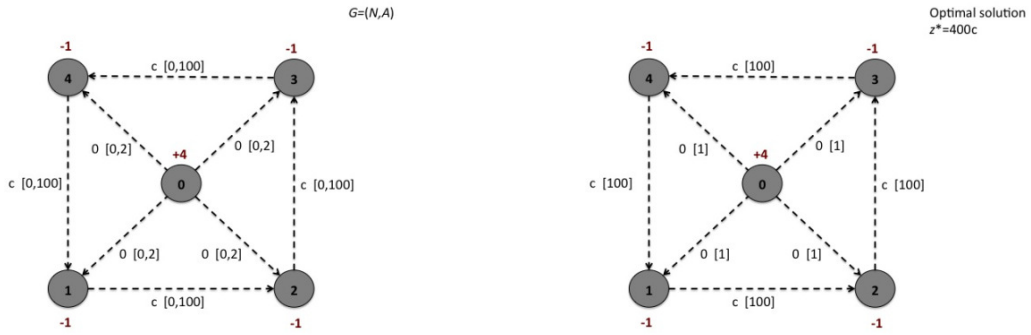

Figure 3 Original network $G = (V, E)$ and optimal solution.

Residual network $G(\mathbf{x}^0)$**.** The initial feasible vertex $\mathbf{x}^0 = (1, 1, 1, 1, 0, 0, 0, 0)^\intercal$ of cost $z^0 = 0$ sends one unit on each radial arc and none on the ring ones. The solution $\mathbf{x}^0$ is basic, where the basic variables are the free variables, $x^0_{0j} = 1,\ j = 1, \ldots, 4$, strictly within the interval $[0, 2]$ on G. Figure 4(left) depicts the residual network $G(\mathbf{x}^0)$, a representation of the residual problem $RP(\mathbf{x}^0)$ (4) defined in terms of the nonnegative $\vec{\mathbf{y}}, \overleftarrow{\mathbf{y}}$-variables. For $j = 1, \ldots, 4$, the free arcs y_{0j} and y_{j0} have a null cost while their residual upper bounds are equal to 1; otherwise for the ring arcs (variables y_{12}, y_{23}, y_{34}, and y_{41}), the cost is c and the residual upper bounds are equal to 100. Figure 4(right) also shows the residual network $G(\mathbf{x}^\star)$ for the optimal solution $\mathbf{x}^\star$ exhibited in Figure 3.

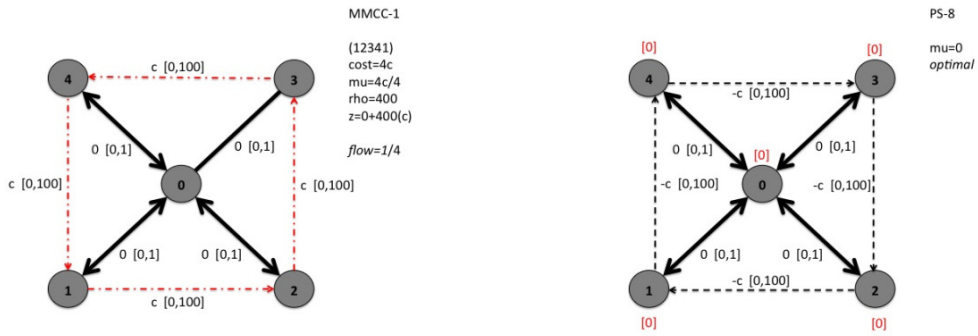

Figure 4 Residual networks $G(\mathbf{x}^0)$ and $G(\mathbf{x}^\star)$.

MMCC. Due to the flow conservation equations on a network flow problem, a feasible

convex combination of variables solution of the residual problem $RP(\mathbf{x}^0)$ (4) corresponds to a *directed cycle* in $G(\mathbf{x}^0)$. Disregarding the residual upper bounds, the arc representation of these cycles, that is, $(\vec{\mathbf{y}}, \bar{\mathbf{y}}) \in \mathbb{R}_+^{|E|+f}$, are *extreme rays* for the system of flow conservation equations (4) on $G(\mathbf{x}^0)$. Let ω denote such a directed cycle with equal flows on the $|\omega|$ selected arcs, that is, a flow of $1/|\omega|$ unit on every arc. Therefore, finding the minimum reduced cost μ on $G(\mathbf{x}^0)$ reduces to evaluating the minimum *mean cost* of directed cycles. The residual network $G(\mathbf{x}^0)$ comprises 13 directed cycles (this number excludes the 2-node cycles). That is, for $j = 1, \ldots, 4$, the 3-node cycles denoted $\mathcal{C}_3^j$ of mean cost $c/3$ represented by the node sequences (0120), (0230), (0340) and (0410), the 4-node cycles $\mathcal{C}_4^j$ of mean cost $2c/4$ given by (01230), (02340), (03410) and (04120), the 5-node cycles $\mathcal{C}_5^j$ of mean cost $3c/5$ given by (012340), (023410), (034120) and (041230), and finally the *outer* ring cycle $\mathcal{C}_o = (12341)$ of mean cost $4c/4 = c$, indeed the minimum value for μ. On the left-hand side of Figure 4, MMCC selects this cycle $\mathcal{C}_o$, where the flow is equal to $y_{12} = y_{23} = y_{34} = y_{41} = 1/4$. The step size in the system (15) must therefore satisfy $\rho(1/4) \leq 100$ on these ring arcs, hence $\rho = 400$, $z^1 = z^0 + 400(c)$, and MMCC reaches the optimal vertex $\mathbf{x}^\star$ in a single iteration. At this point, $G(\mathbf{x}^\star)$ on the right-hand side of Figure 4 contains no negative cost directed cycle, hence satisfies the primal optimality condition of Proposition 1.

PS and IPS. Iterations of PS and IPS can also be seen as the selection of cycles on successive residual networks, with directed cycles for IPS but with some undirected cycles for PS. Indeed, IPS only hides the free variables while searching for the minimum mean cost cycle of the pricing problem whereas PS hides all basic variables. At the first iteration, both algorithms select any one of the 3-node cycles and this induces a nondegenerate pivot. Figure 5 shows the selection of directed cycle (1201) with $y_{12} = 1$ for both algorithms and the objective function becomes $z^1 = c$. For PS, arc (1, 2) enters the basis with flow value 1 while arc (2, 0) is removed from it. For IPS, arc (1, 2) enters the basis with a step size $\rho = 1$ and becomes free while arcs (2, 0) and (0, 1) reach their residual upper bounds. At the second iteration of IPS, the algorithm selects the directed cycle (230412) of cost $3c$ averaged over nonfree arcs (2, 3) and (4, 1) only, that is, $y_{23} = y_{41} = 1/2$, $\mu = 3c/2$, and $\rho(1/2) \leq 1$. Objective function becomes $z^2 = z^1 + 2(3c/2) = 4c$. PS is quite different from IPS at this second iteration. As PS hides the basic arcs, one does not know their orientation while searching for the minimum mean cycle cost, the average being always computed over a single arc, see residual problem (20) where row size is $s + 1 = 1$. PS selects *undirected* cycle (23012) with entering variable $y_{23} = 1$, hence $\mu = 2c$. *A posteriori* computing the impact on the basic variables, one finds that $\rho\, y_{01} \leq 0$ must be satisfied with $y_{01} = 1$, hence a degenerate pivot for PS. Indeed, the column vector associated with y_{23} can be written as a combination of the basic columns with positive, null, and negative values that we are only aware when computing the impact on these.

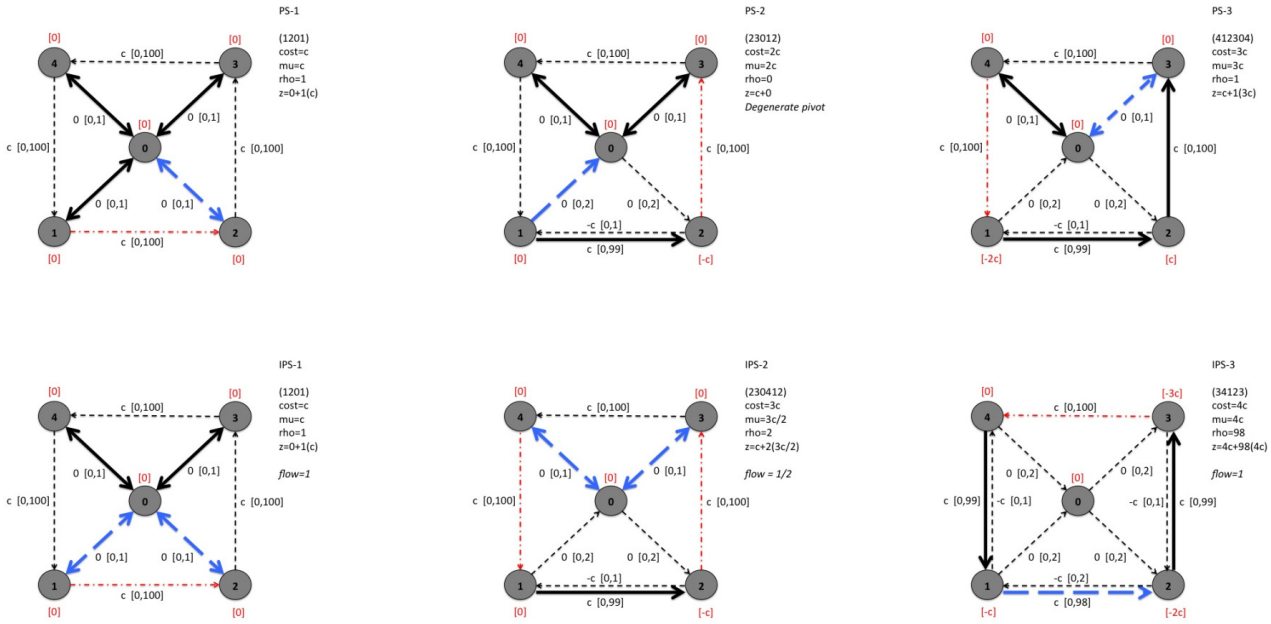

Figure 5 Successive residual networks for PS and IPS (first 3 iterations).

Oscillations of μ. Table 2 provides the iterative processes for the three algorithms. Given the selected ω-cycles, one finds that PS performs seven iterations, one pivot being degenerate, to reach the optimal vertex $\mathbf{x}^\star$ whereas IPS needs five nondegenerate pivots. This means that the single interior direction selected by MMCC can be seen as six successive edge directions selected by PS or five ones selected by IPS. Note that PS always uses a unit flow on a selected cycle as there is only one visible arc for any cycle. For IPS, the flow value also depends on the number of visible arcs (the non free variables) while MMCC counts all arcs of the selected cycle. Finally, observe the oscillations of μ for both PS and IPS. The behavior of μ is well studied in the case of MMCC and is used to show the strongly polynomial complexity of the algorithm, see Goldberg and Tarjan (1989), Radzik and Goldberg (1994), and Gauthier et al. (2014b).

	Primal Simplex							**Improved Primal Simplex**					**MMCC**
k	1	2	3	4	5	6	7	1	2	3	4	5	1
ω-cycle	(1201)	(23012)	(412304)	(34123)	(02340)	(0340)	(1041)	(1201)	(230412)	(34123)	(10341)	(40234)	(12341)
cost c_ω	c	$2c$	$3c$	$4c$	$2c$	c	c	c	$3c$	$4c$	$2c$	$3c$	$4c$
flow	1	1	1	1	1	1	1	1	1/2	1	1/2	1	1/4
μ	c	$2c$	$3c$	$4c$	$2c$	c	c	c	$3c/2$	$4c$	c	$2c$	c
ρ	1	0	1	98	1	1	1	1	2	98	2	1	400
z	c	c	$4c$	$396c$	$398c$	$399c$	$400c$	c	$4c$	$396c$	$398c$	$400c$	$400c$

Table 2 Iterative processes for PS, IPS and MMCC.

Aggregated pricing networks. Figure 6 presents successive network representations of the pricing problems of PS and IPS for the first three iterations. In PS, the pricing problem hides the spanning tree made of all the basic arcs, hence aggregates the network to a single node. PS' pricing problem always sees exactly $|V| - f = 4$ cycles, the number of nonbasic variables for which we have indicated the reduced cost. Some of the cycles induce zero length step size as the one selected at iteration 2.

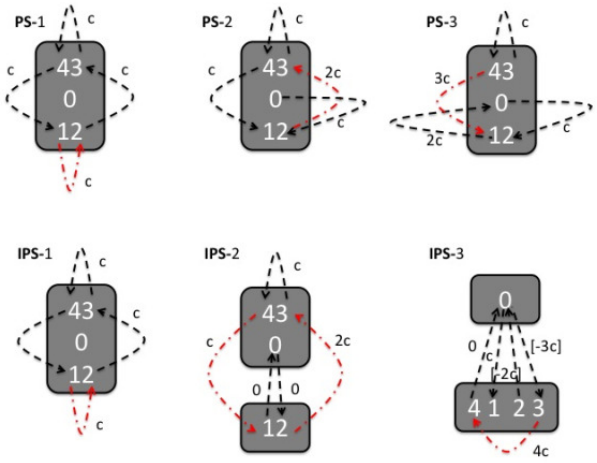

Figure 6 Network flow pricing problems of PS and IPS (first 3 iterations).

This is similar with IPS: free arcs form a forest of trees, each tree being represented by a single (aggregated) node. The directed path between the entering and exiting original nodes of an aggregated node is known *a posteriori* by using the appropriate free arcs. Partial reduced cost is given next to a visible arc. IPS only sees cycles with positive step size and this is why there are only three cycles at iteration 2, one less than that for PS. As there is no aggregation performed on the residual network for MMCC, this algorithm sees all the directed cycles. This can be observed on Figure 4(right) at iteration 1 where there are 13 directed cycles on $G(\mathbf{x}^0)$ compared to four for PS and IPS.

Interior directions. Finally, Table 3 shows direction $\mathbf{y} = \begin{bmatrix} \bar{\mathbf{y}}_F - \bar{\mathbf{y}}_F \\ \bar{\mathbf{y}}_L \end{bmatrix}$ induced by the selected directed cycle on $G(\mathbf{x}^0)$. The first four are edge directions as they are those generated by IPS. Although the nine other directed cycles are not combinations of the 3-node cycles on $G(\mathbf{x}^0)$, it is easy to verify that the corresponding induced directions are combinations of the edge directions, as stated in Proposition 7.

$\mathbf{y}$/*cycle*	$\mathcal{C}_3^1$	$\mathcal{C}_3^2$	$\mathcal{C}_3^3$	$\mathcal{C}_3^4$	$\mathcal{C}_4^1$	$\mathcal{C}_4^2$	$\mathcal{C}_4^3$	$\mathcal{C}_4^4$	$\mathcal{C}_5^1$	$\mathcal{C}_5^2$	$\mathcal{C}_5^3$	$\mathcal{C}_5^4$	$\mathcal{C}_o$
$y_{01} - y_{10}$	1			-1	1		-1		1	-1			
$y_{02} - y_{20}$	-1	1				1		-1		1	-1		
$y_{03} - y_{30}$		-1	1		-1		1				1	-1	
$y_{04} - y_{40}$			-1	1		-1		1	-1			1	
y_{12}	1				1			1	1		1	1	1
y_{23}		1			1	1			1	1		1	1
y_{34}			1			1	1		1	1	1		1
y_{41}				1			1	1		1	1	1	1
	Edge directions				*Interior directions*								

Table 3 Direction $\mathbf{y}$ induced by the selected directed cycle on $G(\mathbf{x}^0)$.

5. Conclusion

This paper unites under one generic framework several known algorithms with a broad spectrum of possibilities whereby both extreme cases correspond to the Primal Simplex algorithm and the Minimum Mean Cycle-Canceling algorithm. Several properties are established for different family members. In particular, a family that provides nondegenerate pivots at every iteration and another one for which the pricing problems provide edge directions only. IPS is the only variant which qualifies for both features.

While interior directions are certainly usual in the realm of nonlinear optimization, it is not so often that one thinks about such possibilities for simplex-like algorithms. The oracle for such findings remains linear and does not require fancy derivatives.

It is often noted in column generation that giving more structure to the pricing problem allows faster retrieval of improving columns. A potentially benefical avenue might therefore be the use of *dual-optimal inequalities*. The idea is to add to the primal formulation some additional variables and columns for which the corresponding dual inequalities are always satisfied by an optimal dual solution. These have been used in several applications such as the cutting stock and the bin packing problems, see Valério de Carvalho (2005) and Ben Amor et al. (2006).

While MMCC is a strongly polynomial algorithm for networks, we believe it is worthwhile to investigate whether IPS applied on this family of problems is also strongly polynomial. It is true that the observed oscillations of μ causes difficulty in understanding its behavior but Gauthier et al. (2014b) have discarded the importance of the iteration-wise behavior of μ and rather emphasize the analysis on so-called phases, a group of iterations satisfying certain properties. In the spirit of this renewed interest towards this particular algorithm, a linear programming derivative is a research path that deserves some attention.

All in all, the implementation of this generic framework is a chapter of its own. While this paper has no computational study pretension, several ideas have already been suggested as promising assets. The inverse $\mathbf{T}^{-1}$ induces structure in the technological matrix whereas the residual problem automates degeneracy screening. By combining both constructions, extensive variable screening in the pricing step can be done on several fronts such as compatibility and partial reduced costs to residual upper bounds ratios. The first kind

being easy performed pivots whereas the second kind aims at detecting good cost to step pivots.

Acknowledgments

Jacques Desrosiers acknowledges the National Science and Engineering Research Council of Canada for its financial support.

References

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ, USA, 1993.

Hatem M. T. Ben Amor, Jacques Desrosiers, and José M. Valério de Carvalho. Dual-Optimal Inequalities for Stabilized Column Generation. *Operations Research*, 54(3):454–463, 2006. doi:10.1287/opre.1060.0278.

Hatem M. T. Ben Amor, Jacques Desrosiers, and Antonio Frangioni. On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167–1184, 2009. doi:10.1016/j.dam.2008.06.021.

Pascal Benchimol, Guy Desaulniers, and Jacques Desrosiers. Stabilized dynamic constraint aggregation for solving set partitioning problems. *European Journal of Operational Research*, 223 (2):360–371, 2012. doi:10.1016/j.ejor.2012.07.004.

George B. Dantzig and Mukund N. Thapa. *Linear Programming 1: Introduction*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, USA, 1997.

George B. Dantzig and Mukund N. Thapa. *Linear Programming 2: Theory and Extensions*. Springer Series in Operations Research and Financial Engineering (Book 2). Springer, New York, NY, USA, 2003.

George B. Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960. doi:10.1287/opre.8.1.101.

Guy Desaulniers, Jacques Desrosiers, Irina Ioachim, Marius M. Solomon, François Soumis, and Daniel Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In TeodorGabriel Crainic and Gilbert Laporte, editors, *Fleet Management and Logistics*, pages 57–93. Springer, New York, NY, USA, 1998. doi:10.1007/978-1-4615-5755-5_3.

Jacques Desrosiers, Yvan Dumas, Marius M. Solomon, and François Soumis. Time constrained routing and scheduling. In Michael Ball, Tom Magnanti, Clyde Monma, and George Nemhauser, editors, *Handbooks in Operations Research and Management Science, Vol. 8: Network Routing*, volume 8, chapter 2, pages 35–139. Elsevier, Maryland Heights, MO, USA, October 1995. doi:10.1016/S0927-0507(05)80106-9.

Olivier du Merle, Daniel Villeneuve, Jacques Desrosiers, and Pierre Hansen. Stabilized column generation. *Discrete Mathematics*, 194:229–237, 1999. doi:10.1016/S0012-365X(98)00213-1.

Issmail Elhallaoui, Daniel Villeneuve, François Soumis, and Guy Desaulniers. Dynamic aggregation of set partitioning constraints in column generation. *Operations Research*, 53(4):632–645, 2005. doi:10.1287/opre.1050.0222.

Issmail Elhallaoui, Guy Desaulniers, Abdelmoutalib Metrane, and François Soumis. Bi-dynamic constraint aggregation and subproblem reduction. *Computers & Operations Research*, 35(5): 1713–1724, 2008. doi:10.1016/j.cor.2006.10.007.

Issmail Elhallaoui, Abdelmoutalib Metrane, Guy Desaulniers, and François Soumis. An Improved Primal Simplex algorithm for degenerate linear programs. *INFORMS Journal on Computing*, 23:569–577, 2011. doi:10.1287/ijoc.1100.0425.

Jean Bertrand Gauthier, Jacques Desrosiers, and Marco E. Lübbecke. Decomposition theorems for linear programs. *Operations Research Letters*, 42(8):553–557, December 2014a. doi:10.1016/j.orl.2014.10.001.

Jean Bertrand Gauthier, Jacques Desrosiers, and Marco E. Lübbecke. About the minimum mean cycle-canceling algorithm. *Discrete Applied Mathematics*, 2014b. doi:10.1016/j.dam.2014.07.005.

Jean Bertrand Gauthier, Jacques Desrosiers, and Marco E. Lübbecke. Tools for primal degenerate linear programs: IPS, DCA, and PE. *EURO Journal on Transportation and Logistics*, 2015. doi:10.1007/s13676-015-0077-5.

Andrew V. Goldberg and Robert Endre Tarjan. Finding minimum-cost circulations by canceling negative cycles. *Journal of the ACM*, 36(4):873–886, 1989. doi:10.1145/76359.76368.

Edward Seymour Klotz. *Dynamic Pricing Criteria in Linear Programming*. PhD thesis, Stanford University, Stanford, CA, USA, July 1988.

Marco E. Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005. doi:10.1287/opre.1050.0234.

Abdelmoutalib Metrane, François Soumis, and Issmail Elhallaoui. Column generation decomposition with the degenerate constraints in the subproblem. *European Journal of Operational Research*, 207(1):37–44, 2010. doi:10.1016/j.ejor.2010.05.002.

Tomasz Radzik and Andrew V. Goldberg. Tight bounds on the number of minimum-mean cycle cancellations and related results. *Algorithmica*, 11(3):226–242, 1994. doi:10.1007/BF01240734.

Vincent Raymond, François Soumis, Abdelmoutalib Metrane, and Jacques Desrosiers. Positive edge: A pricing criterion for the identification of non-degenerate simplex pivots. Les Cahiers du GERAD G-2010-61, HEC Montréal, Montreal, QC, Canada, 2010.

Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., Chichester, West Sussex, England, 1986.

Tamás Terlaky and Shuzhong Zhang. Pivot rules for linear programming: A survey on recent theoretical developments. *Annals of Operations Research - Annals OR*, 46-47(1):203–233, 1993. doi:10.1007/BF02096264.

Mehdi Towhidi, Jacques Desrosiers, and François Soumis. The positive edge criterion within COIN-OR's CLP. *Computers & Operations Research*, 49(0):41–46, 2014. doi:10.1016/j.cor.2014.03.020.

José M. Valério de Carvalho. Using extra dual cuts to accelerate convergence in column generation. *INFORMS Journal on Computing*, 17(2), 2005. doi:10.1287/ijoc.1030.0060.