**Les Cahiers du GERAD** ISSN: 0711–2440

# Tools for Primal Degenerate Linear Programs

J.B. Gauthier, J. Desrosiers,
M.E. Lübbecke

G–2013–51

September 2013

# Tools for Primal Degenerate Linear Programs

**Jean Bertrand Gauthier**
**Jacques Desrosiers**

*GERAD & HEC Montréal*
*Montréal (Québec) Canada, H3T 2A7*

jean-bertrand.gauthier@hec.ca
jacques.desrosiers@hec.ca

**Marco E. Lübbecke**

*RWTH Aachen University, Operations Research*
*D-52072 Aachen, Germany*

marco.luebbecke@rwth-aachen.de

September 2013

*Les Cahiers du GERAD*

G–2013–51

**Abstract:** This paper describes three recent tools for dealing with primal degeneracy in linear programming. The first one is the *Improved Primal Simplex* algorithm (IPS) which turns degeneracy into a possible advantage. The constraints of the original problem are dynamically partitioned based on the numerical values of the current basic variables. The idea is to work only with those constraints that correspond to non-degenerate basic variables. This leads to a row-reduced problem which decreases the size of the current working basis. The main feature of IPS is that it provides a non-degenerate pivot at every iteration of the solution process until optimality is reached. To achieve such a result, a negative reduced cost convex combination of non-basic variables is selected such that it is compatible with the current row-reduced problem, if any. This pricing step provides a necessary and sufficient optimality condition for linear programming.

The second tool is the *Dynamic Constraint Aggregation* (DCA), a constructive strategy specifically designed for set partitioning constraints. It heuristically aims to achieve the properties provided by the IPS methodology. We bridge the similarities and differences of IPS and DCA on set partitioning models. The final tool is the *Positive Edge* rule (PE). It capitalizes on the compatibility definition to determine the variable status during the reduced cost computation. Within IPS, this added value is obtained without explicitly computing the updated column components in the simplex-tableau. Since the selection of a compatible variable to enter the basis ensures a non-degenerate pivot, PE permits a trade-off between strict improvement and high reduced cost degenerate pivots. Ultimately, we establish tight bonds between these three tools by going back to the linear algebra framework from which emanates the so-called concept of vector subspace basis.

**Key Words:** Primal simplex, degeneracy, row-reduced problem, combination of entering variables, positive edge rule, non-degenerate pivot algorithm, Dantzig-Wolfe decomposition, vector subspace.

**Acknowledgments:** Jacques Desrosiers acknowledges the Natural Sciences and Engineering Research Council of Canada for its financial support.

# 1 Introduction

In the primal simplex algorithm (see Dantzig, 1963), degeneracy comes in two flavors: degenerate solutions and degenerate pivots. The former is a question of observation, it is a dichotomous state of the solution which either exhibits degenerate basic variables or not. Geometrically speaking, a degenerate solution corresponds to an overrepresented vertex meaning that several equivalent bases are associated with the same basic solution. The latter is the culprit of the algorithm in more ways than one. In fact, it is the only phenomenon which actually jeopardizes the convergence of the simplex algorithm both theoretically and empirically. Degeneracy questions the efficiency of the simplex algorithm and creates ambiguity in the post-analysis. On the one hand, degeneracy can affect the efficiency of the algorithm in obtaining an optimal solution because it creates redundant work. More specifically, a degenerate pivot amounts to trading one degenerate basic variable for a non-basic one. It is in the aftermath of the computations that one ultimately realizes the wasted effort. On the other hand, a byproduct of the simplex method is the sensitivity analysis done after the optimization. Each constraint is associated with a dual variable whose value depends on the chosen basis. Since an optimal degenerate basis is not uniquely defined, it can mislead the interpretation of two otherwise equivalent solutions.

Degeneracy has been under scrutiny for practically as long as linear programming. We distinguish two lines of studies from the literature. The first aims to eliminate degeneracy all together and the other provides guidelines to alleviate its impact. On the first count, think of the work of Charnes (1952) which revolves around modifying the polytope of the whole solution space in such a way that no two solutions ever share the same vertex. The concept amounts to right-hand side perturbations thus creating slight variations in the way the hyperplanes intersect. While the idea of eradicating degeneracy altogether is appealing, today's simplex codes use a more *ad hoc* strategy which sends us to the second count. The contributions of Wolfe (1963) and Ryan and Osborne (1988) are abundant evidence that applying this strategy as necessary is highly effective. The perturbations are now applied in an adaptive manner and on a more local scale. *Stabilization* extends the idea of perturbation by incorporating dual information. Penalty functions, trust regions and expansion strategies are among the instrumental concepts of stabilization as described in the papers of du Merle et al. (1999) and Ben Amor et al. (2009). Column generation benefits from the concept as it tackles the particular sensitivity to the values of dual variables during the resolution process.

Numerous pivot rules have also been proposed to avoid performing degenerate pivots. In this regards, the work of Terlaky and Zhang (1993) is enlightening in many respects. Indeed, while many of these rules share common properties and sometimes even correspond to special cases of one another, they are distinguished according to certain properties: feasibility maintenance, anti-cycling feature and recursive nature. While there might have been hope about the performance of many of these rules, even non degenerate instances can be difficult to optimize as supported by Klee and Minty (1972). The performance of a pivot rule may therefore be considered as a trade-off between its complexity and the savings it procures with respect to the number of iterations. The state of the art in terms of degenerate problems seems to be the Devex rule of Harris (1973). We underline that regardless of their intricacies, all of these rules have a limited gain with respect to the absence of guaranteed efficiency. That is, zero-step length pivots could still ensue from the chosen direction. The anti-cycling feature present in Bland (1977) or Fukuda (1982) ensures this behavior does not happen indefinitely, however it is generally accepted that taking expensive measures to protect against cycling is not worthwhile.

A new trend appears in the late nineties with the paper of Pan (1998) which formulates a generic basis for degenerate solutions. Embedding this concept in a column generation scheme led to the *Dynamic Constraint Aggregation* (DCA) of Elhallaoui et al. (2005) for the set partitioning problem. This problem landed itself particularly well to such a concept because of its peculiar structure. Indeed, it is this very structure that allows DCA to heuristically harness the power of a generic basis and find strictly improving pivots for intermediate integer solutions. The paper of Elhallaoui et al. (2011) extends the algorithmic methodology with the *Improved Primal Simplex*. As its name would have it, this extension takes place with regards to any linear programming problem at every iteration. In a nut shell, the structure of solutions is preemptively taken into account in order to drive the next simplex pivot in a strictly improving direction. That structure is dynamically updated with respect to the current solutions.

Degeneracy is a phenomenon encountered particularly often for linear programming relaxations of combinatorial optimization problems. Set partitioning and set covering models are prominent examples of practical relevance: Vehicle routing and crew scheduling problems (and many related problems in transportation and logistics) are most successfully formulated this way (Desrosiers et al., 1995; Desaulniers et al., 1998). In vehicle routing, it is immediate how degeneracy occurs: Constraints represent (often large numbers) of tasks to be covered by relatively few numbers of vehicles or crew members, that is, only few variables assume a positive value in an integer solution. It should be noted that column generation to solve linear programs with a huge number of variables (Barnhart et al., 1998) is a natural extension of the primal simplex, and as such suffers from degeneracy as well.

This methodological paper describes three related tools for dealing with primal degeneracy. It first summarizes various results on the *Improved Primal Simplex* algorithm (IPS). It is based on the original papers of Elhallaoui et al. (2011), Raymond et al. (2009, 2010b), Metrane et al. (2010) and its generalization to row-reduced column generation (Desrosiers et al., 2013b). The second tool is the *Dynamic Constraint Aggregation* method (DCA) of Elhallaoui et al. (2005, 2008). This is a constructive strategy specifically designed for set partitioning constraints that heuristically aims to achieve the properties provided by the IPS methodology. We bridge the similarities and differences of these two methods on set partitioning models. Finally, at the heart of IPS and DCA lies the search for compatible entities. While manipulating the compatible set can be computationally efficient, identifying said set can be time consuming for large problems. The *Positive Edge* rule (PE) (Raymond et al., 2010a; Towhidi et al., 2012) aims to simplify this verification by extracting the compatibility status during the reduced cost computation. A compatibility-test can be done in linear time using the original column-vector, $O(m)$, where $m$ is the number of constraints.

The paper is organized as follows. Section 2 first exposes the theory of IPS with several hints to the primal simplex algorithm. By casting the linear algebra framework on our study, Sect. 3 presents another perspective of IPS. Section 4 addresses the more practical side with regards to several implementation choices. The importance of compatibility in the design of specialized applications is highlighted in Sect. 5. The similarities and differences between IPS and DCA are examined in Sect. 6 while Sect. 7 reveals the PE rule. Various results from the literature are reported at the end of Sect. 2, Sect. 6 and Sect. 7 depending on the context of the underlying tool. Our conclusions end the paper in Sect. 8.

**Notation.** Vectors and matrices are written in bold face. We denote by $\mathbf{I}_\ell$ the $\ell \times \ell$ identity matrix and by $\mathbf{0}$ (resp. $\mathbf{1}$) a vector/matrix with all zeros (resp. ones) entries of appropriate contextual dimensions. For a subset $I \subseteq \{1, \ldots, m\}$ of row-indices and a subset $J \subseteq \{1, \ldots, n\}$ of column-indices, we denote by $\mathbf{A}_{IJ}$ the sub-matrix of $\mathbf{A}$ containing the rows and columns indexed by $I$ and $J$, respectively. We further use standard linear programming notation like $\mathbf{A}_J\mathbf{x}_J$, the subset of columns of $\mathbf{A}$ indexed by $J$ multiplied by the corresponding sub-vector of variables $\mathbf{x}_J$. There is one notable exception: the set $N$ does *not* denote the non-basis but rather the set of degenerate variables.

## 2 Improved Primal Simplex

This section first exposes the theory of IPS in the context of a linear program with lower and upper bounded variables. Essentially, we provide a change of variables, the construction of a generic basis and the resulting row partition, the column partition in terms of compatible and incompatible variables, and the development of a pricing subproblem. In Sect. 2.3, we also establish the improvement step from a solution $\mathbf{x}^0$ to the next $\mathbf{x}^1$ which incidentally brings an inspiring twist to the pivoting rule. Sect. 2.4 provides the proof of a necessary and sufficient optimality condition derived from the pricing step. Sect. 2.5 presents a simplified version of IPS for linear programs in standard form. For a better understanding of the concepts, an illustrative example is given in Sect. 2.6 on a small linear program.

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

$$\begin{array}{rlll} z^\star := \min & \mathbf{c}^\intercal \mathbf{x} & & \\ \text{s.t.} & \mathbf{A}\mathbf{x} & = \mathbf{b}, & [\boldsymbol{\pi}] \\ \mathbf{0} \leq \mathbf{l} \leq & \mathbf{x} \leq \mathbf{u}, & & \end{array} \tag{1}$$

where $\mathbf{x}, \mathbf{c}, \mathbf{l}, \mathbf{u} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^m \times \mathbb{R}^n$, and $m < n$. We assume that $\mathbf{x}$ includes $m$ artificial variables, hence $\mathbf{A} := (\mathbf{a}_j)_{j \in \{1,\dots,n\}}$ is a matrix of full row-rank. In presence of these artificial variables, LP is feasible and bounded. Finally, $\boldsymbol{\pi} \in \mathbb{R}^m$ is a vector of dual variables associated with the equality constraints.

Let $\mathbf{x}^0$, represented by $(\mathbf{x}^0_F; \mathbf{x}^0_L; \mathbf{x}^0_U)$, be a feasible solution where the three sub-vectors are defined according to the value of their variables: $\mathbf{x}^0_L$ at their lower bounds, $\mathbf{x}^0_U$ at their upper bounds, and non-degenerate variables $\mathbf{l}_F < \mathbf{x}^0_F < \mathbf{u}_F$. Let there be $f := |F|$ such *free* variables, $0 \le f \le m$, that can take values above or below the current ones. Partition matrix $\mathbf{A} = [\mathbf{A}_F, \mathbf{A}_L, \mathbf{A}_U]$ and cost vector $\mathbf{c}^\intercal = [\mathbf{c}^\intercal_F, \mathbf{c}^\intercal_L, \mathbf{c}^\intercal_U]$ accordingly. Perform a change of variables on the degenerate ones:

$$\begin{array}{lll} \mathbf{x}_F := & \mathbf{x}_F & \\ \mathbf{x}_L := & \mathbf{x}^0_L + \mathbf{y}_L, & \mathbf{y}_L \ge \mathbf{0} \\ \mathbf{x}_U := & \mathbf{x}^0_U - \mathbf{y}_U, & \mathbf{y}_U \ge \mathbf{0}. \end{array} \tag{2}$$

Let $N := L \cup U$ to form $\mathbf{y}_N = (\mathbf{y}_L; \mathbf{y}_U)$. This vector is bounded above by $\mathbf{r}_N$, where $r_j := u_j - \ell_j, \forall j \in N$. Let $\mathbf{d}^\intercal_N := [\mathbf{c}^\intercal_L, -\mathbf{c}^\intercal_U]$ and define $\mathbf{A}^0_N := [\mathbf{A}_L, -\mathbf{A}_U]$, that is, $\mathbf{a}^0_j = \mathbf{a}_j,\ \forall j \in L$, and $\mathbf{a}^0_j = -\mathbf{a}_j,\ \forall j \in U$. Given the adjusted right-hand side $\mathbf{b}^0 := \mathbf{b} - \mathbf{A}_L\mathbf{x}^0_L - \mathbf{A}_U\mathbf{x}^0_U$, LP becomes:

$$\begin{array}{rlcccl} z^\star := \mathbf{c}^\intercal_L\mathbf{x}^0_L + \mathbf{c}^\intercal_U\mathbf{x}^0_U + & \min & \mathbf{c}^\intercal_F\mathbf{x}_F & + & \mathbf{d}^\intercal_N\mathbf{y}_N & \\ & \text{s.t.} & \mathbf{A}_F\mathbf{x}_F & + & \mathbf{A}^0_N\mathbf{y}_N & = \mathbf{b}^0, \quad [\boldsymbol{\pi}] \\ & & \mathbf{l}_F \le \mathbf{x}_F \le \mathbf{u}_F, & & \mathbf{0} \le \mathbf{y}_N \le \mathbf{r}_N. & \end{array} \tag{3}$$

The main idea in IPS is to reduce the number of constraints from $m$ to $f$, the number of free variables in the current solution. The advantage of this row-reduction is a smaller working basis of dimension $f \times f$ rather than the usual larger one of dimension $m \times m$. This will come at the expense of a more involved pricing step which solves a linear program of row-size $m - f + 1$ to select an improving subset of columns, that is, a convex combination of columns with two properties: it must be compatible with the current row-reduced problem (see Definition 1) and its convex combination of reduced costs is negative. If such a combination exists, a strict improvement in the objective function value occurs, otherwise the current solution is optimal.

## 2.1 A generic basis and the row partition

Assume $f < m$ and that the columns of $\mathbf{A}_F$ are linearly independent. This is the case if $\mathbf{x}^0$ comes from the solution of a simplex-type algorithm. The proposed methodology also applies for $f = m$. Indeed, there is no row-reduction but the current solution in non-degenerate, and so is the pivot.

We construct a *generic* $m \times m$ basis matrix $\mathbf{B}$ which represents any and all combinations of degenerate variables which may complete the basis. Such a generic basis is readily available using the $f$ free variables associated with the columns of $\mathbf{A}_F$ together with $m - f$ artificial variables taking value zero. The selection of an appropriate set of artificial variables can be done by solving a primal simplex phase I problem over columns of $\mathbf{A}_F$ and those of the identity matrix $\mathbf{I}_m$ with the corresponding vector of artificial variables here denoted $\boldsymbol{\lambda}$:

$$\begin{array}{llcll} \min & & & \mathbf{1}^\intercal\boldsymbol{\lambda} & \\ \text{s.t.} & \mathbf{A}_F\mathbf{x}_F & + & \mathbf{I}_m\boldsymbol{\lambda} & = \mathbf{b}^0, \\ & \mathbf{x}_F \ge \mathbf{0}, & & \boldsymbol{\lambda} \ge \mathbf{0}. & \end{array} \tag{4}$$

Solving this problem is undoubtedly successful in accordance with the fact that $\mathbf{A}_F\mathbf{x}^0_F = \mathbf{b}^0$. Let it be clear that this construction process must identify some subset $\mathbf{A}_{PF}$ of exactly $f$ independent rows from matrix $\mathbf{A}_F$. Basis $\mathbf{B}$ is one of the many bases available to identify the overrepresented vertex $\mathbf{x}^0$. As such, observe the sensitivity of the dual vector $\boldsymbol{\pi}^\intercal := \mathbf{c}^\intercal_B\mathbf{B}^{-1}$ with respect to the choice of basis completion.

**Row partition.** Let $\bar{\mathbf{b}}^0 := \mathbf{B}^{-1}\mathbf{b}^0$. Constraints of LP are divided according to the actual values of $\bar{\mathbf{b}}^0$: for row-set $P$, $\bar{\mathbf{b}}^0_P > \mathbf{0}$; for the remaining rows in set $Z$, $\bar{\mathbf{b}}^0_Z = \mathbf{0}$. Basis $\mathbf{B}$ and its inverse $\mathbf{B}^{-1}$ are as follows:

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_{PF} & \mathbf{0} \\ \mathbf{A}_{ZF} & \mathbf{I}_{m-f} \end{bmatrix} \text{ and } \mathbf{B}^{-1} = \begin{bmatrix} \mathbf{A}^{-1}_{PF} & \mathbf{0} \\ -\mathbf{A}_{ZF}\mathbf{A}^{-1}_{PF} & \mathbf{I}_{m-f} \end{bmatrix}, \tag{5}$$

where matrix $\mathbf{A}_{PF}$ of dimension $f \times f$ is the working basis. LP becomes

$$\begin{array}{rrcrcll}
z^\star := \mathbf{c}_L^\intercal \mathbf{x}_L^0 + \mathbf{c}_U^\intercal \mathbf{x}_U^0 + \min & \mathbf{c}_F^\intercal \mathbf{x}_F & + & \mathbf{d}_N^\intercal \mathbf{y}_N & & & \\
\text{s.t.} & \mathbf{A}_{PF}\mathbf{x}_F & + & \mathbf{A}_{PN}^0 \mathbf{y}_N & = & \mathbf{b}_P^0, & [\boldsymbol{\pi}_P] \\
& \mathbf{A}_{ZF}\mathbf{x}_F & + & \mathbf{A}_{ZN}^0 \mathbf{y}_N & = & \mathbf{b}_Z^0, & [\boldsymbol{\pi}_Z] \\
& \multicolumn{2}{r}{\mathbf{l}_F \le \mathbf{x}_F \le \mathbf{u}_F,} & \multicolumn{3}{l}{\mathbf{0} \le \mathbf{y}_N \le \mathbf{r}_N.} &
\end{array} \tag{6}$$

The current solution is given by $\mathbf{x}_F^0 = \mathbf{A}_{PF}^{-1}\mathbf{b}_P^0 = \bar{\mathbf{b}}_P^0$ while $\mathbf{y}_N^0 = \mathbf{0}$. Let

$$\bar{\mathbf{A}}_N^0 := \mathbf{B}^{-1}\mathbf{A}_N^0 = \begin{bmatrix} \bar{\mathbf{A}}_{PN}^0 \\ \bar{\mathbf{A}}_{ZN}^0 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{PF}^{-1}\mathbf{A}_{PN}^0 \\ \mathbf{A}_{ZN}^0 - \mathbf{A}_{ZF}\mathbf{A}_{PF}^{-1}\mathbf{A}_{PN}^0 \end{bmatrix}. \tag{7}$$

The new LP formulation obtained after the change of variables and the left-multiplication by $\mathbf{B}^{-1}$ of the set of equality constraints makes degeneracy more evident:

$$\begin{array}{rrcrcll}
z^\star := \mathbf{c}_L^\intercal \mathbf{x}_L^0 + \mathbf{c}_U^\intercal \mathbf{x}_U^0 + \min & \mathbf{c}_F^\intercal \mathbf{x}_F & + & \mathbf{d}_N^\intercal \mathbf{y}_N & & & \\
\text{s.t.} & \mathbf{x}_F & + & \bar{\mathbf{A}}_{PN}^0 \mathbf{y}_N & = & \bar{\mathbf{b}}_P^0, & [\boldsymbol{\psi}_P] \\
& & & \bar{\mathbf{A}}_{ZN}^0 \mathbf{y}_N & = & \mathbf{0}, & [\boldsymbol{\psi}_Z] \\
& \multicolumn{2}{r}{\mathbf{l}_F \le \mathbf{x}_F \le \mathbf{u}_F,} & \multicolumn{3}{l}{\mathbf{0} \le \mathbf{y}_N \le \mathbf{r}_N.} &
\end{array} \tag{8}$$

Dual vector $\boldsymbol{\pi}$ can be retrieved from above dual vector $\boldsymbol{\psi}$ using the expression $\boldsymbol{\pi}^\intercal = \boldsymbol{\psi}^\intercal \mathbf{B}^{-1}$:

$$\boldsymbol{\pi}_P^\intercal = \boldsymbol{\psi}_P^\intercal \mathbf{A}_{PF}^{-1} - \boldsymbol{\psi}_Z^\intercal \mathbf{A}_{ZF}\mathbf{A}_{PF}^{-1} \tag{9}$$

$$\boldsymbol{\pi}_Z^\intercal = \boldsymbol{\psi}_Z^\intercal. \tag{10}$$

Observe that any solution to (8) must satisfy $\bar{\mathbf{A}}_{ZN}^0 \mathbf{y}_N = \mathbf{0}$. Non-degenerate variables $\mathbf{x}_F = \bar{\mathbf{b}}_P^0$ are basic in row-set $P$, hence reduced cost vector $\bar{\mathbf{c}}_F = \mathbf{c}_F - \boldsymbol{\psi}_P = \mathbf{0}$. Therefore $\boldsymbol{\psi}_P = \mathbf{c}_F$, but $\boldsymbol{\psi}_Z$ is yet undetermined. The current solution $\mathbf{x}^0 = (\mathbf{x}_F^0; \mathbf{x}_L^0; \mathbf{x}_U^0)$ for (1), or equivalently $(\mathbf{x}_F^0; \mathbf{y}_N^0 = \mathbf{0})$ for (8), is optimal if there exists some dual vector $\boldsymbol{\psi}_Z$ such that

$$\bar{d}_j := d_j - \mathbf{c}_F^\intercal \bar{\mathbf{a}}_{Pj}^0 - \boldsymbol{\psi}_Z^\intercal \bar{\mathbf{a}}_{Zj}^0 \ge 0, \ \forall j \in N.$$

## 2.2 The pricing step

Let $\mu := \min_{j \in N} \bar{d}_j$ be the smallest reduced cost value for $\mathbf{y}_N$ given $\boldsymbol{\psi}_P = \mathbf{c}_F$ but optimized over $\boldsymbol{\psi}_Z$. Finding $\mu$ can be formulated as a linear program:

$$\begin{array}{lllll}
\max & \mu & & & \\
\text{s.t.} & \mu & \le & d_j - \mathbf{c}_F^\intercal \bar{\mathbf{a}}_{Pj}^0 - \boldsymbol{\psi}_Z^\intercal \bar{\mathbf{a}}_{Zj}^0, & \forall j \in N.
\end{array} \tag{11}$$

Let $\tilde{d}_j := d_j - \mathbf{c}_F^\intercal \bar{\mathbf{a}}_{Pj}^0,\ j \in N$, be the *partial reduced cost* of variable $y_j$ computed by only using the known dual vector $\boldsymbol{\psi}_P = \mathbf{c}_F$, or equivalently $\tilde{\mathbf{d}}_N^\intercal := \mathbf{d}_N^\intercal - \mathbf{c}_F^\intercal \bar{\mathbf{A}}_{PN}^0$ in vector form. Therefore

$$\begin{array}{lllllll}
\max & \mu & & & & & \\
\text{s.t.} & \mathbf{1}\mu & + & \boldsymbol{\psi}_Z^\intercal \bar{\mathbf{A}}_{ZN}^0 & \le & \tilde{\mathbf{d}}_N, & [\mathbf{y}_N]
\end{array} \tag{12}$$

where $\mathbf{y}_N \ge \mathbf{0}$ is the associated vector of dual variables for the inequality constraints. Taking the dual of (12), the pricing problem is written in terms of $\mathbf{y}_N$, the vector of null variables to price out:

$$\begin{array}{rlrcll}
\mu := & \min & \tilde{\mathbf{d}}_N^\intercal \mathbf{y}_N & & & \\
& \text{s.t.} & \mathbf{1}^\intercal \mathbf{y}_N & = & 1, & [\mu] \\
& & \bar{\mathbf{A}}_{ZN}^0 \mathbf{y}_N & = & \mathbf{0}, & [\boldsymbol{\psi}_Z] \\
& & \mathbf{y}_N & \ge & \mathbf{0}. &
\end{array} \tag{13}$$

Solution $\mathbf{x}^0$ is optimal if $\mu \geq 0$. Otherwise an optimal solution $\mathbf{y}_N^0 = (\mathbf{y}_L^0; \mathbf{y}_U^0)$ to (13) identifies a convex combination of variables such that $\bar{\mathbf{A}}_{ZN}^0 \mathbf{y}_N^0 = \mathbf{0}$. This pricing problem (13) can be solved by the dual simplex algorithm because only the convexity constraint is not satisfied by the current value $\mathbf{y}_N = \mathbf{0}$. Alternatively, specialized algorithms can be used in some applications. This is the case for LP defined as a capacitated minimum cost network flow problem where pricing problem (13) corresponds to a minimum mean cost cycle problem which can be solved in $O(mn)$ time by dynamic programming (Karp, 1978).

## 2.3 The exchange mechanism: from $\mathbf{x}^0$ to an improved solution $\mathbf{x}^1$

When $\mu < 0$, we consider two cases for the solution $\mathbf{y}_N^0$ obtained from the pricing problem: the optimal convex combination only contains a single positive variable or, for the general case, it involves one or more positive variables. Regardless of the solution, observe that the pricing problem finds only an improving direction of negative reduced cost value $\mu$, if one exists. The step size $\rho$ is governed by the usual pivot rule in the simplex-tableau formulation (8).

**Special case:** $y_j^0 = 1,\ j \in N$.
In that case, $\bar{\mathbf{a}}_{Zj}^0 = \mathbf{0}$ and $\mu = \tilde{d}_j$. The ratio-test is performed on row-set $P$ only, using $\bar{\mathbf{a}}_{Pj}^0$ and $\bar{\mathbf{b}}_P^0 > \mathbf{0}$ and we have the following definition for a vector $\mathbf{a} \in \mathbb{R}^m$:

**Definition 1** $\mathbf{a} \in \mathbb{R}^m$ *is* compatible with row-set P *if and only if* $\bar{\mathbf{a}}_Z := \mathbf{a}_Z - \mathbf{A}_{ZF}\mathbf{A}_{PF}^{-1}\mathbf{a}_P = \mathbf{0}$.

Therefore, column $\mathbf{a}_j^0,\ j \in N$, enters basis $\mathbf{B}$, one column leaves it, and the step size $\rho$ is computed according to the maximum increase of variable $y_j$, that is, $y_j = \rho y_j^0 = \rho$. From (8), we have the following relations:

$$\begin{array}{ccccc} \mathbf{x}_F & + & \bar{\mathbf{a}}_{Pj}^0 \rho & = & \bar{\mathbf{b}}_P^0 \\ \mathbf{l}_F \leq \mathbf{x}_F \leq \mathbf{u}_F, & & 0 \leq \rho \leq r_j. & & \end{array} \tag{14}$$

Let $\bar{\mathbf{a}}_{Pj}^0 := (\bar{a}_{ij}^0)_{i\in P}$. Step-size $\rho$ can increase up to the upper bound $r_j$, or according to the maximum change in the vector of free variables $\mathbf{l}_F \leq \mathbf{x}_F = \bar{\mathbf{b}}_P^0 - \bar{\mathbf{a}}_{Pj}^0 y_j \leq \mathbf{u}_F$, where $P$ and $F$ are in a one-to-one correspondence upon a reordering of the free variables:

$$\rho := \min \left\{ r_j, \min_{i\in P|\bar{a}_{ij}^0>0} \left\{ \frac{\bar{b}_i^0 - l_i}{\bar{a}_{ij}^0} \right\}, \min_{i\in P|\bar{a}_{ij}^0<0} \left\{ \frac{u_i - \bar{b}_i^0}{-\bar{a}_{ij}^0} \right\} \right\} > 0\ . \tag{15}$$

The vector of free variables becomes $\mathbf{x}_F = \bar{\mathbf{b}}_P^0 - \rho\ \bar{\mathbf{a}}_{Pj}^0$, where $\bar{\mathbf{b}}_P^0 = \mathbf{x}_F^0$. From (8), the change in the objective function value comes from the following:

$$\begin{aligned} \mathbf{c}_F^{\intercal}\mathbf{x}_F + d_j\rho &= \mathbf{c}_F^{\intercal}(\mathbf{x}_F^0 - \rho\ \bar{\mathbf{a}}_{Pj}^0) + d_j\rho \\ &= \mathbf{c}_F^{\intercal}\mathbf{x}_F^0 + \rho(d_j - \mathbf{c}_F^{\intercal}\mathbf{x}_F^0) = \mathbf{c}_F^{\intercal}\mathbf{x}_F^0 + \rho\tilde{d}_j. \end{aligned} \tag{16}$$

Hence $z$ improves by $\Delta z = \rho\tilde{d}_j = \rho\mu$. Either $j \in L$ ($x_j$ is at its lower bound) or $j \in U$ ($x_j$ is at its upper bound). Therefore, the $\mathbf{x}^0$-solution is updated to $\mathbf{x}^1$ according to the change of variables in (2) and where $y_j^0 = 1$ appears only once, either within $\mathbf{y}_L^0$ or $\mathbf{y}_U^0$:

$$\begin{array}{lcl} \mathbf{x}_F^1 := \mathbf{x}_F^0 & - & \rho\,\bar{\mathbf{a}}_{Pj}^0 \\ \mathbf{x}_L^1 := \mathbf{x}_L^0 & + & \rho\,\mathbf{y}_L^0 \\ \mathbf{x}_U^1 := \mathbf{x}_U^0 & - & \rho\,\mathbf{y}_U^0\ . \end{array} \tag{17}$$

The number of free variables in the new $\mathbf{x}^1$-solution is at most $f$, that is, the new solution can be more degenerate. If $\rho < r_j$, $f$ decreases if more than one of the free variables reach their bounds. Otherwise $\rho = r_j$, the corresponding $x_j$ variable changes bound and therefore stays degenerate in the new solution; the number of free variables decreases if at least one free variable reaches a bound.

**General case:** $\mathbf{y}_N^0$ is composed of one or several positive variables.
The number of positive variables is at most $m - f + 1$, the row dimension of the pricing problem. Vector $\bar{\mathbf{A}}_{ZN}^0 \mathbf{y}_N^0 = \mathbf{0}$, that is, the convex combination of columns of $\bar{\mathbf{A}}_{ZN}^0$ results in a null vector in row-set $Z$. Hence, by Definition 1, $\mathbf{A}_N^0 \mathbf{y}_N^0 \in \mathbb{R}^m$ is *compatible with row-set* $P$ and our analysis follows the steps already described in the above special case.

Column $\mathbf{A}_N^0 \mathbf{y}_N^0$ enters basis $\mathbf{B}$ while one leaves it. Denote by $\theta_\omega, \omega \in \Omega$, the entering variable, a new type of variables indexed by convex combination $\omega$ if $\mathbf{y}_N^0$ is composed of more than one positive variable. Parameters of $\theta_\omega$ relative to formulation (8) are as follows: $\bar{\mathbf{a}}_w^0 = \begin{bmatrix} \bar{\mathbf{a}}_{Pw}^0 \\ \bar{\mathbf{a}}_{Zw}^0 \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}}_{PN}^0 \mathbf{y}_N^0 \\ \mathbf{0} \end{bmatrix}$, reduced cost $\mu$, cost $\mathbf{d}_N^\intercal \mathbf{y}_N^0$, and $\mathbf{y}_N^0 \neq \mathbf{0}$. With the addition of variable $\theta_\omega$ to the LP model in (8), we have the following relations, where relevant parameters are indicated within brackets:

$$\begin{array}{ccccc} \mathbf{x}_F & + & [\bar{\mathbf{A}}_{PN}^0 \mathbf{y}_N^0]\ \theta_\omega & = & \bar{\mathbf{b}}_P^0 \\ \mathbf{l}_F \leq \mathbf{x}_F \leq \mathbf{u}_F, & & 0 \leq [\mathbf{y}_N^0]\ \theta_\omega \leq \mathbf{r}_N . & & \end{array} \tag{18}$$

In the above expressions, $[\mathbf{y}_N^0]\theta_\omega \leq \mathbf{r}_N$ can also be written as $[y_j^0]\theta_\omega \leq r_j,\ \forall j \in N$. Therefore, the step size $\rho$ for the entering variable $\theta_\omega$ is computed as

$$\rho := \min \left\{ \min_{j \in N | y_j^0 > 0} \left\{ \frac{r_j}{y_j^0} \right\}, \min_{i \in P | \bar{a}_{iw}^0 > 0} \left\{ \frac{\bar{b}_i^0 - l_i}{\bar{a}_{iw}^0} \right\}, \min_{i \in P | \bar{a}_{iw}^0 < 0} \left\{ \frac{u_i - \bar{b}_i^0}{-\bar{a}_{iw}^0} \right\} \right\}. \tag{19}$$

Once again a non-degenerate pivot occurs ($\rho > 0$) and the objective function strictly improves by

$$\Delta z = \rho \mu = \rho\, \tilde{\mathbf{d}}_N^\intercal \mathbf{y}_N^0\ . \tag{20}$$

The $\mathbf{x}^0$-solution is updated to $\mathbf{x}^1$ according to the change of variables:

$$\begin{array}{lcl} \mathbf{x}_F^1 := \mathbf{x}_F^0 & - & \rho\, \bar{\mathbf{a}}_{P\omega}^0 \\ \mathbf{x}_L^1 := \mathbf{x}_L^0 & + & \rho\, \mathbf{y}_L^0 \\ \mathbf{x}_U^1 := \mathbf{x}_U^0 & - & \rho\, \mathbf{y}_U^0\ . \end{array} \tag{21}$$

The number of free variables in $\mathbf{x}^1$ is at most $f + (m - f + 1) - 1 = m$, that is, the new solution can be more degenerate but it can also be less degenerate when several degenerate variables of the convex combination become free.

Regardless of the manner in which one updates the current solution, the aftermath is the result of an *exchange mechanism*. Even the ratio-test performed to identify the exiting variable in the traditional primal simplex echoes this notion. Indeed, the exchange always happens in a one-to-one fashion, while we have just seen that it can be more involved. Given the current solution, the exchange mechanism starts in the pricing problem (13) for rows in set $Z$ by finding in $\bar{\mathbf{A}}_{ZN}^0 \mathbf{y}_N = \mathbf{0},\ \mathbf{y}_N \geq \mathbf{0}$, directions $\mathbf{y}_L^0$ and $-\mathbf{y}_U^0$ for vectors $\mathbf{x}_L$ and $\mathbf{x}_U$, respectively. The exchange process is afterward completed by using rows in set $P$ and interval constraints in (18): direction for vector $\mathbf{x}_F$ is given by $-\bar{\mathbf{A}}_{PN}^0 \mathbf{y}_N^0$ and the step-size is derived in expression (19). In the latter, it occurs between $\mathbf{x}_F$ and entering variable $\theta_\omega, \omega \in \Omega$, either selected as a single compatible variable or as an appropriate combination of a number of incompatible ones.

## 2.4 A characterization of linear programming optimality

In summary, when $\mu \geq 0$, the current solution $\mathbf{x}^0$ is optimal. Otherwise, $\mu < 0$ and we obtain a strict improvement of the objective function, update the current solution from $\mathbf{x}^0$ to $\mathbf{x}^1$, and the process is repeated until the following *necessary and sufficient optimality condition* is met.

**Proposition 1** *A basic feasible solution* $\mathbf{x}^0 = (\mathbf{x}_F^0; \mathbf{x}_L^0; \mathbf{x}_U^0)$ *is an optimal solution to the linear program* (1) *if and only if there exists a dual vector* $\boldsymbol{\psi}_Z$ *such that* $\mu \geq 0$*, as optimized by the primal-dual pair* (12)–(13) *of the pricing problem.*

**Proof.** Formulations (1) and (8) are equivalent. Because $\bar{\mathbf{c}}_F = \mathbf{0}$, if there exists some dual vector $\boldsymbol{\psi}_Z$ such that $\mathbf{d}_N^\intercal - \mathbf{c}_F^\intercal \bar{\mathbf{A}}_{PN}^0 - \boldsymbol{\psi}_Z^\intercal \bar{\mathbf{A}}_{ZN}^0 \geq \mathbf{0}^\intercal$, $N := L \cup U$, then $(\bar{\mathbf{c}}_F, \bar{\mathbf{d}}_N) \geq \mathbf{0}$. Therefore, $\boldsymbol{\psi}^\intercal = (\mathbf{c}_F^\intercal, \boldsymbol{\psi}_Z^\intercal)$ provides a feasible dual solution to (8). Since $\boldsymbol{\psi}_P^\intercal \bar{\mathbf{b}}_P^0 = \mathbf{c}_F^\intercal \mathbf{x}_F^0$, primal and dual objective functions are equal and the current feasible solution $\mathbf{x}^0$ is optimal for (1).

To show the converse, let $\mathbf{x}^0$ be an optimal solution to (1) and assume $\mu < 0$. An optimal solution to the pricing problem (13) identifies a convex combination of variables such that a non-degenerate pivot occurs ($\rho > 0$) and the objective function strictly improves by $\rho\mu < 0$. This contradicts the optimality of $\mathbf{x}^0$ and completes the proof. □

All simplex derivatives work according to the presumption of innocence. Optimality is indeed assumed until proven otherwise. It is no different in IPS, yet it is an amazing feat that the content of the pricing problem be reminiscent of the no more, no less punch line. The sufficient condition answers to the first part, while the necessary condition to the second.

## 2.5 IPS for a linear program in standard form

The reader may recall that incorporating lower and upper bounds in the primal simplex method adds a plethora of intricacies in the algorithmic analysis. Although the same is true of IPS, we assumed the reader was sufficiently accustomed with the primal simplex method. In the spirit of conveying the general idea of IPS, it might be worthwhile to present a simpler version. This basically amounts to removing the dimension $U$ from the formulation. The simplifications are threesome and correspond to the main steps of IPS: creating the column and row partitions, building the pricing problem, and modifying the current solution. Given LP in standard form

$$\begin{array}{rll} z^\star := \min & \mathbf{c}^\intercal \mathbf{x} & \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}, & [\boldsymbol{\pi}] \\ & \mathbf{x} \geq \mathbf{0}, & \end{array} \tag{22}$$

and a feasible solution $\mathbf{x}^0 = (\mathbf{x}_F^0; \mathbf{x}_N^0)$, the column partition step distinguishes between currently non-degenerate (or positive or free) variables $\mathbf{x}_F^0$ and degenerate (or null) variables $\mathbf{x}_N^0$:

$$\begin{array}{rccccl} z^\star := \min & \mathbf{c}_F^\intercal \mathbf{x}_F & + \mathbf{c}_N^\intercal \mathbf{x}_N & & & \\ \text{s.t.} & \mathbf{A}_F \mathbf{x}_F & + \mathbf{A}_N \mathbf{x}_N & = & \mathbf{b}, & [\boldsymbol{\pi}] \\ & \mathbf{x}_F \geq \mathbf{0}, & \mathbf{x}_N \geq \mathbf{0}. & & & \end{array} \tag{23}$$

Recall the previous change of variables in (2). It should be clear that $\mathbf{x}_N := \mathbf{y}_N$, $\mathbf{A}_N^0 = \mathbf{A}_N$, and $\mathbf{b}^0 = \mathbf{b}$.

$$\begin{array}{rcccccl} z^\star := \min & \mathbf{c}_F^\intercal \mathbf{x}_F & + & \mathbf{c}_N^\intercal \mathbf{y}_N & & & \\ \text{s.t.} & \mathbf{x}_F & + & \bar{\mathbf{A}}_{PN} \mathbf{y}_N & = & \bar{\mathbf{b}}_P, & [\boldsymbol{\psi}_P] \\ & & & \bar{\mathbf{A}}_{ZN} \mathbf{y}_N & = & \mathbf{0}, & [\boldsymbol{\psi}_Z] \\ & \mathbf{x}_F \geq \mathbf{0}, & & \mathbf{y}_N \geq \mathbf{0}. & & & \end{array} \tag{24}$$

Once again, the linear transformations performed on the original system underline the degeneracy of the current solution. Furthermore, any solution must satisfy $\bar{\mathbf{A}}_{ZN} \mathbf{y}_N = \mathbf{0}$ in (24). Therefore, the pricing problem can be written in terms of the vector of null variables to price out, and the current partial reduced cost vector $\tilde{\mathbf{c}}_N^\intercal := \mathbf{c}_N^\intercal - \boldsymbol{\psi}_P^\intercal \bar{\mathbf{A}}_{PN} = \mathbf{c}_N^\intercal - \mathbf{c}_F^\intercal \bar{\mathbf{A}}_{PN}$:

$$\begin{array}{rrcll} \mu := \min & \tilde{\mathbf{c}}_N^\intercal \mathbf{y}_N & & & \\ \text{s.t.} & \mathbf{1}^\intercal \mathbf{y}_N & = & 1, & [\mu] \\ & \bar{\mathbf{A}}_{ZN} \mathbf{y}_N & = & \mathbf{0}, & [\boldsymbol{\psi}_Z] \\ & \mathbf{y}_N & \geq & \mathbf{0}. & \end{array} \tag{25}$$

Solution $\mathbf{x}^0 = (\mathbf{x}_F^0; \mathbf{x}_N^0)$ is optimal for LP in (22) if $\mu \geq 0$. Otherwise $\mu < 0$ and an optimal solution $\mathbf{y}_N^0$ to (25) identifies a convex combination of variables such that $\bar{\mathbf{A}}_{ZN} \mathbf{y}_N^0 = \mathbf{0}$. The convex combination established by the pricing problem may once again contain one or several $\mathbf{y}$-variables. Let $\theta_\omega, \omega \in \Omega$, be the entering variable with the following parameters: reduced cost $\mu$, cost $\mathbf{c}_N^\intercal \mathbf{y}_N^0$, and $\bar{\mathbf{a}}_\omega = \begin{bmatrix} \bar{\mathbf{a}}_{P\omega} \\ \bar{\mathbf{a}}_{Z\omega} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}}_{PN}\mathbf{y}_N^0 \\ \mathbf{0} \end{bmatrix}$. What matters is that the ratio-test (19) is now computed with a single component:

$$\rho := \min_{i \in P | \bar{a}_{i\omega} > 0} \left\{ \frac{\bar{b}_i}{\bar{a}_{i\omega}} \right\} > 0. \tag{26}$$

A non-degenerate pivot occurs and the objective of LP in (22) strictly improves by $\Delta z = \rho\,\mu$. Finally, the $\mathbf{x}^0$-solution is updated to $\mathbf{x}^1$ as follows:

$$\begin{aligned} \mathbf{x}_F^1 &:= \mathbf{x}_F^0 - \rho\,\bar{\mathbf{a}}_{P\omega} \\ \mathbf{x}_N^1 &:= \rho\,\mathbf{y}_N^0. \end{aligned} \tag{27}$$

## 2.6 A numerical example

Table 1 depicts a linear program in standard form comprising eight $\mathbf{x}$-variables and six constraints. Degenerate solution $\mathbf{x}^0$ is already presented in the simplex-tableau format: $(x_1^0, x_2^0, x_3^0) = (30, 25, 50)$ are the positive (or free) basic variables and the basis has been completed with artificial $\boldsymbol{\lambda}$-variables in rows 4, 5, and 6. The cost of this solution is $z^0 = 185$.

Table 1: Simplex-tableau at $\mathbf{x}^0$

| | $x_1$ | $x_2$ | $x_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | | | $\theta_\omega$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{c}$ | 2 | 3 | 1 | | | | 10 | 17 | -20 | 14 | -4 | | | -5 |
| | 1 | | | | | | 2 | 2 | 1 | -5 | 7 | = | 30 | 1 |
| | | 1 | | | | | 4 | 3 | -5 | 10 | -10 | = | 25 | -2 |
| | | | 1 | | | | -3 | 1 | 2 | 3 | 11 | = | 50 | 5 |
| | | | | 1 | | | 0 | 0 | 6 | 5 | -13 | = | 0 | 0 |
| | | | | | 1 | | 0 | 0 | 3 | 4 | -8 | = | 0 | 0 |
| | | | | | | 1 | 0 | 0 | 3 | -4 | 0 | = | 0 | 0 |
| $\mathbf{x}^0$ | 30 | 25 | 50 | | | | | | | | | $z^0 =$ | 185 | |
| $\tilde{\mathbf{c}}$ | | | | | | | -3 | 3 | -9 | -9 | 1 | | | -6 |

Dual vector $\mathbf{c}_F^\intercal = [2, 3, 1]$ is used for computing partial reduced cost vector $\tilde{\mathbf{c}}_N^\intercal$ = [-3, 3, -9, -9, 1]. By inspection, we see that $x_4$ and $x_5$ are compatible with the row partition derived from the right-hand side values. One can observe that the associated columns are (trivial) combinations of the (unit) vectors of the free variables $x_1, x_2$ and $x_3$. Indeed, *compatible vectors* belong to the vector space spanned by the free variable vectors, see Proposition 2.

Both compatible variables would provide a non degenerate simplex pivot if chosen as entering variables but only $x_4$ has a negative partial reduced cost value $\tilde{c}_4$ = -3 (which is also equal to its reduced cost $\bar{c}_4$). Incompatible variables $x_6$ and $x_7$ possess the same negative partial reduced cost value of -9 whereas $\tilde{c}_8 = 1$. The selection of incompatible variable $x_6$ or $x_7$ would result in a degenerate simplex pivot while that of $x_8$ would increase the objective function by $1 \times (\frac{30}{7})$.

However, solving the pricing problem (25) over the last three rows results in a combination of the incompatible vectors with weigths: $(y_6^0, y_7^0, y_8^0) = (0.4, 0.3, 0.3)$. This provides the compatible vector $\bar{\mathbf{a}}_\omega^\intercal$ = [1 -2 5 0 0 0] for variable $\theta_\omega$ of reduced cost $\mu$ = -9(0.4) - 9(0.3) + 1(0.3) = -6 and cost -5. Ratio-test on the top three rows results in $\rho = \min\{\frac{30}{1}, -, \frac{50}{5}\} = 10$. Entering variable $\theta_\omega$ provides a strict improvement of the objective function of -6 $\times$ 10 = -60. Variable $\theta_\omega$ takes value 10, $x_3$ goes out of the basis, and other free variables $x_1$ and $x_2$ are respectively updated to 20 and 45. Alternatively, variables $x_6, x_7$ and $x_8$ can be entered one by one in the basis, in any order, and this provides the same results. In the new solution of cost 125, the positive variables are $(x_1, x_2, \theta_\omega) = (20, 45, 10)$ or equivalently for $\mathbf{x}^1$ in terms of the original variables, $(x_1, x_2, x_6, x_7, x_8) = (20, 45, 4, 3, 3)$ while $x_3, x_4$ and $x_5$ are null variables.

From the five columns corresponding to the positive variables, the first five rows are independent and artificial variable $\lambda_6$ is basic at value zero in the last row. Inverse basis $\mathbf{B}^{-1}$ at $\mathbf{x}^1$ appears in Table 2 and is used to construct the next degenerate simplex-tableau in Table 3.

Table 2: Basis $\mathbf{B}$ and its inverse at $\mathbf{x}^1$

$$\mathbf{B} = \left[\begin{array}{ccccc|c} 1 & & 1 & -5 & 7 & \\ & 1 & -5 & 10 & -10 & \\ & & 2 & 3 & 11 & \\ & & 6 & 5 & -13 & \\ & & 3 & 4 & -8 & \\ \hline & & 3 & -4 & 0 & 1 \end{array}\right] \quad \mathbf{B}^{-1} = \left[\begin{array}{ccccc|c} 1 & & -0.20 & -2.133 & 4.067 & \\ & 1 & 0.40 & 5.600 & -9.800 & \\ & & 0.08 & 0.453 & -0.627 & \\ & & 0.06 & -0.327 & 0.613 & \\ & & 0.06 & 0.007 & -0.053 & \\ \hline 0 & 0 & 0 & -2.667 & 4.333 & 1 \end{array}\right]$$

Table 3: Simplex-tableau at $\mathbf{x}^1$

| | $x_1$ | $x_2$ | $x_6$ | $x_7$ | $x_8$ | $\lambda_6$ | $x_3$ | $x_4$ | $x_5$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{c}$ | 2 | 3 | -20 | 14 | -4 | | 1 | 10 | 17 | | |
| | 1 | | | | | | -0.20 | 2.60 | 1.80 | = | 20 |
| | | 1 | | | | | 0.40 | 2.80 | 3.40 | = | 45 |
| | | | 1 | | | | 0.08 | -0.24 | 0.08 | = | 4 |
| | | | | 1 | | | 0.06 | -0.18 | 0.06 | = | 3 |
| | | | | | 1 | | 0.06 | -0.18 | 0.06 | = | 3 |
| | | | | | | 1 | 0 | 0 | 0 | = | 0 |
| $\mathbf{x}^1$ | 20 | 45 | 4 | 3 | 3 | | | | | $z^1 =$ | 125 |
| $\tilde{\mathbf{c}}$ | | | | | | | 1.2 | -6.6 | 4.2 | | |

$\mathbf{A}_{PF}^{-1}$, the inverse of the working basis within $\mathbf{B}^{-1}$, is used to compute the row-set $P$ dual variable vector $\mathbf{c}_F^\intercal \mathbf{A}_{PF}^{-1} = [2, 3, \text{-}0.2, \text{-}1.133, 0.067]$ and partial reduced costs $(\tilde{c}_3, \tilde{c}_4, \tilde{c}_5) = (1.2, \text{-}6.6, 4.2)$. Moreover,

$$\bar{\mathbf{a}}_{Zj} := -\mathbf{A}_{ZF}\mathbf{A}_{PF}^{-1}\mathbf{a}_{Pj} + \mathbf{a}_{Zj} = \mathbf{0},\ j \in \{3, 4, 5\},$$

characterizes column compatibility by computing

$$\left[\ \bar{\mathbf{a}}_{Z3}\ \bar{\mathbf{a}}_{Z4}\ \bar{\mathbf{a}}_{Z5}\ \right] = (0,\ 0,\ 0,\ \text{-}2.667,\ 4.333)\left[\ \mathbf{a}_{P3}\ \mathbf{a}_{P4}\ \mathbf{a}_{P5}\ \right] + \left[\ 0\ \ 0\ \ 0\ \right] = \left[\ 0\ \ 0\ \ 0\ \right].$$

Null variables $x_3$, $x_4$ and $x_5$ are compatible with the current row partition, and the optimal solution to the pricing problem at iteration 1 is $y_4^1 = 1$. Variable $x_4$ enters the basis, being the only one with a negative reduced cost of -6.6. Ratio-test on top five rows results in $\rho = \min\{\frac{20}{2.6}, \frac{45}{2.8}, -, -, -\} = 7.692$ and entering variable $x_4$ provides an objective function improvement of -6.6 × 7.692 = -50.769. Variable $x_1$ goes out of the basis, and updated free variables $x_2, x_6, x_7$ and $x_8$ appear in Table 4, here presented in terms of the simplex-tableau at $\mathbf{x}^2$ before being updated. Observe that the actual combination of variables $x_6$, $x_7$ and $x_8$ satisfies the last three rows at zero right-hand side. The cost of this solution is $z^2 = 74.231$.

Table 4: Simplex-tableau at $\mathbf{x}^2$ before being updated

| | $x_4$ | $x_2$ | $x_6$ | $x_7$ | $x_8$ | $\lambda_6$ | $x_3$ | $x_1$ | $x_5$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{c}$ | 10 | 3 | -20 | 14 | -4 | | 1 | 2 | 17 | | |
| | 2 | | 1 | -5 | 7 | | | 1 | 2 | = | 30 |
| | 4 | 1 | -5 | 10 | -10 | | | | 3 | = | 25 |
| | -3 | | 2 | 3 | 11 | | 1 | | 1 | = | 50 |
| | | | 6 | 5 | -13 | | | | | = | 0 |
| | | | 3 | 4 | -8 | | | | | = | 0 |
| | | | 3 | -4 | 0 | 1 | | | | = | 0 |
| $\mathbf{x}^2$ | 7.692 | 23.462 | 5.846 | 4.385 | 4.385 | | | | | $z^2 =$ | 74.231 |
| $\tilde{\mathbf{c}}$ | | | | | | | 0.692 | 2.538 | 8.769 | | |

$\mathbf{B}^{-1}$ at $\mathbf{x}^2$ appears in Table 5 from which we compute $\mathbf{c}_F^\intercal \mathbf{A}_{PF}^{-1} = [-0.538,\ 3,\ 0.308,\ 4.282,\ -10.256]$ and partial reduced costs $(\tilde{c}_3, \tilde{c}_1, \tilde{c}_5) = (0.692, 2.538, 8.769)$. Since these are positive, $\mathbf{x}^2$ is optimal.

Table 5: Inverse basis $\mathbf{B}^{-1}$ at $\mathbf{x}^2$

$$\mathbf{B}^{-1} = \left[\begin{array}{rrrrr|r} 0.385 & & \text{-}0.077 & \text{-}0.821 & 1.564 & \\ \text{-}1.077 & 1 & 0.615 & 7.897 & \text{-}14.179 & \\ 0.092 & & 0.062 & 0.256 & \text{-}0.251 & \\ 0.069 & & 0.046 & \text{-}0.474 & 0.895 & \\ 0.069 & & 0.046 & \text{-}0.141 & 0.228 & \\ \hline 0 & 0 & 0 & \text{-}2.667 & 4.333 & 1 \end{array}\right]$$

## 2.7 Computational results

A comparison between IPS and the primal simplex algorithm of `CPLEX` was performed for linear programs in standard form. At a given iteration of IPS, degenerate variables compatible with the current row partition are identified by an external procedure explicitly using inverse $\mathbf{A}_{PF}^{-1}$, i.e., those in set $C \subseteq N$ such that $\bar{\mathbf{A}}_{ZC} = \mathbf{0}$. Row-reduced problem

$$\begin{array}{lccccc} \min & \mathbf{c}_F^\intercal \mathbf{x}_F & + & \mathbf{c}_C^\intercal \mathbf{y}_C & & \\ \text{s.t.} & \mathbf{A}_{PF}\mathbf{x}_F & + & \mathbf{A}_{PC}\mathbf{y}_C & = & \mathbf{b}_P, \\ & \mathbf{x}_F \geq \mathbf{0}, & & \mathbf{y}_C \geq \mathbf{0}, & & \end{array} \tag{28}$$

is solved by the primal simplex over $\mathbf{x}_F$ and $\mathbf{y}_C = \mathbf{x}_C$. Then, the pricing subproblem (25) is solved by the dual simplex over all incompatible variables $\mathbf{y}_I$, $I := N \setminus C$, to select several independent convex combinations, each one with a negative reduced cost. The new row-reduced problem is re-optimized and the row-partition is updated according to the new solution.

On 10 instances involving 2,000 constraints and up to 10,000 variables for *simultaneous vehicle and crew scheduling problems in urban mass transit systems*, IPS reduces CPU times by a factor of 3.53 compared to the primal simplex of `CPLEX` (Elhallaoui et al., 2011; Raymond et al., 2010b). Degeneracy is about 50% for these problems. IPS is also tested on 14 instances of *aircraft fleet assignment*. These consist in maximizing the profits of assigning a type of aircraft to each flight segment over a horizon of one week. The multi-commodity flow formulation comprises 5,000 constraints and 25,000 variables; degeneracy is about 65%. IPS reduces CPU times by a factor of 12.30 on average.

In these fleet assignment problems, an upper bound of 1 can explicitly be imposed on arc flow variables (see also the discussion in Sect. 4.3). Hence, degeneracy occurs for basic variables at 0 or at 1. CPU times are reduced by a factor of 20.23 on average for these LPs with upper bounded variables (Raymond et al., 2009).

# 3 A Linear Algebra Framework

To appreciate the generality of IPS, the reader is invited to consider its presentation only borrows from the algebraic manipulations of the primal simplex algorithm. The linear algebra framework is put forth to derive another way to look at the decomposition. Section 3.1 introduces the vector subspace $\mathbf{V}$ spanned by the column-vectors of $\mathbf{A}_F$. This is followed in Sect. 3.2 by the practical use of an equivalent vector subspace basis $\mathbf{\Lambda}_f$. In Sect. 3.3, we examine a different vector subspace basis, that is, $\mathbf{\Lambda}_r$ of possibly larger dimension $r \geq f$ that is sufficient to span $\mathbf{A}_F$. At the risk of allowing degenerate pivots, this larger vector subspace basis gives a lot of freedom in the implementation and, more importantly, closes the theoretical gap between IPS and DCA.

## 3.1 Vector subspace V

The idea comes down to considering a variable as a vector. All observations made on variables must, by extension, hold from the point of view of vectors. It is therefore possible to associate a variable with a vector in a one-to-one correspondence. Recall that $\mathbf{x}_F$ is the vector of free basic variables.

**Proposition 2** $\mathbf{a} \in \mathbb{R}^m$ *is* compatible with row-set P *if and only if it belongs to* $\mathbf{V}$, *the vector subspace of* $\mathbb{R}^m$ *spanned by* $\mathbf{A}_F$.

**Proof.** Assume vector $\mathbf{a}$ belongs to the vector subspace $\mathbf{V}$ spanned by $\mathbf{A}_F$, that is, for $\boldsymbol{\alpha} \in \mathbb{R}^f$, it can be written as $\mathbf{a} = \mathbf{A}_F\boldsymbol{\alpha}$, or equivalently, $\begin{bmatrix} \mathbf{a}_P \\ \mathbf{a}_Z \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{PF}\boldsymbol{\alpha} \\ \mathbf{A}_{ZF}\boldsymbol{\alpha} \end{bmatrix}$. Since $\mathbf{A}_{PF}$ is non singular, one obtains $\boldsymbol{\alpha} = \mathbf{A}_{PF}^{-1}\mathbf{a}_P$ from the first set of constraints. Substituting in the second set, we have $\mathbf{a}_Z = \mathbf{A}_{ZF}\mathbf{A}_{PF}^{-1}\mathbf{a}_P$ which means that $\bar{\mathbf{a}}_Z := \mathbf{a}_Z - \mathbf{A}_{ZF}\mathbf{A}_{PF}^{-1}\mathbf{a}_P = \mathbf{0}$, hence $\mathbf{a}$ is compatible with row-set P by Definition 1.

To show the converse, let $\mathbf{a}$ be compatible with row-set $P$, that is, $\bar{\mathbf{a}}_Z := \mathbf{a}_Z - \mathbf{A}_{ZF}\mathbf{A}_{PF}^{-1}\mathbf{a}_P = \mathbf{0}$, where $\mathbf{A}_{PF}^{-1}\mathbf{a}_P = \bar{\mathbf{a}}_P$. Hence, $\mathbf{a}_Z = \mathbf{A}_{ZF}\bar{\mathbf{a}}_P$ and $\mathbf{a}_P = \mathbf{A}_{PF}\bar{\mathbf{a}}_P$, or equivalently, $\mathbf{a} = \mathbf{A}_F\bar{\mathbf{a}}_P$. Hence $\mathbf{a}$ belongs to the vector space $\mathbf{V}$. □

Observe that free variables $\mathbf{A}_F$ are compatible whereas degenerate basic variables are not. A consequence of Proposition 2 is that every subset of $f$ independent vectors of $\mathbf{V}$ can be used as a vector subspace basis for $\mathbf{V}$. Let us explicitly recall the definition of a vector basis as a linearly independent spanning set. A simple but important observation is the following: The set of $f$ independent vectors of $\mathbf{A}_F$ identified in IPS is therefore a *minimal* spanning set capable of representing the current solution, $\mathbf{A}_F\mathbf{x}_F^0 = \mathbf{b}^0$. Indeed, the very construction of the working basis in $\mathbf{B}$ implies that $\mathbf{A}_F$ spans $\mathbf{b}^0$, that is, $\mathbf{x}_F^0 = \mathbf{A}_{PF}^{-1}\mathbf{b}_P^0$, see the system of linear equations in (6) or (8).

## 3.2 Vector subspace basis $\boldsymbol{\Lambda}_f$

The identification of the working basis is one of the bottleneck operations of the improved primal simplex algorithm. Furthermore, as the reader can observe from formulation (8), it is useless to multiply by $\mathbf{A}_{PF}^{-1}$ the rows in set $P$ to identify an improving variable $\theta_\omega, \omega \in \Omega$, if any. Indeed, only $\bar{\mathbf{a}}_{P\omega}$ needs to be computed to perform ratio-test (19). An alternative set to $\mathbf{A}_F$ of $f$ linearly independent vectors that spans $\mathbf{V}$ is vector subspace basis $\boldsymbol{\Lambda}_f = \begin{bmatrix} \mathbf{I}_f \\ \mathbf{M} \end{bmatrix}$, where $\mathbf{M} = \mathbf{A}_{ZF}\mathbf{A}_{PF}^{-1}$. Together with $\boldsymbol{\Lambda}_f^\perp = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{m-f} \end{bmatrix}$, it provides the vector basis $\mathbf{T} := [\boldsymbol{\Lambda}_f, \boldsymbol{\Lambda}_f^\perp]$ of $\mathbb{R}^m$ and its inverse $\mathbf{T}^{-1}$:

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_f & \mathbf{0} \\ \mathbf{M} & \mathbf{I}_{m-f} \end{bmatrix} \quad \text{and} \quad \mathbf{T}^{-1} = \begin{bmatrix} \mathbf{I}_f & \mathbf{0} \\ -\mathbf{M} & \mathbf{I}_{m-f} \end{bmatrix}. \tag{29}$$

The LP formulation obtained after the change of variables and the transformation by the more practical $\mathbf{T}^{-1}$ results in an equivalent system for which only the rows in set $Z$ are transformed:

$$\begin{array}{rrcrcll} z^\star := \mathbf{c}_L^\intercal\mathbf{x}_L^0 + \mathbf{c}_U^\intercal\mathbf{x}_U^0 + \min & \mathbf{c}_F^\intercal\mathbf{x}_F & + & \mathbf{d}_N^\intercal\mathbf{y}_N & & & \\ \text{s.t.} & \mathbf{A}_{PF}\mathbf{x}_F & + & \mathbf{A}_{PN}^0\mathbf{y}_N & = & \mathbf{b}_P^0, & [\boldsymbol{\psi}_P] \\ & & & \bar{\mathbf{A}}_{ZN}^0\mathbf{y}_N & = & \mathbf{0}, & [\boldsymbol{\psi}_Z] \\ & \mathbf{l}_F \leq \mathbf{x}_F \leq \mathbf{u}_F, & & \mathbf{0} \leq \mathbf{y}_N \leq \mathbf{r}_N, & & & \end{array} \tag{30}$$

where $\bar{\mathbf{A}}_{ZN}^0 = \mathbf{A}_{ZN}^0 - \mathbf{M}\mathbf{A}_{RN}^0$. Similarly to Equations (9) and (10), dual vector $\boldsymbol{\pi}$ can be retrieved from dual vector $\boldsymbol{\psi}$ in (30) using the expression $\boldsymbol{\pi}^\intercal = \boldsymbol{\psi}^\intercal\mathbf{T}^{-1}$:

$$[\boldsymbol{\pi}_P^\intercal, \boldsymbol{\pi}_Z^\intercal] = \left[\ \boldsymbol{\psi}_P^\intercal - \boldsymbol{\psi}_Z^\intercal\mathbf{M},\ \boldsymbol{\psi}_Z^\intercal\ \right]. \tag{31}$$

When all is said and done, using vector subspace properties enables one to derive a working basis much more efficiently using LU-decomposition. Furthermore, the inverse is implicitly obtained in $\mathbf{M}$ as a byproduct of the decomposition.

## 3.3 Vector subspace basis $\boldsymbol{\Lambda}_r, r \geq f$

Let us consider the more general situation where $r$, the dimension of the vector subspace basis $\boldsymbol{\Lambda}_r$ spanning the current columns of $\mathbf{A}_F$, is larger than or equal to $f$, the number of free variables. Therefore, $\boldsymbol{\Lambda}_r$ includes the $f$ columns of $\mathbf{A}_F$ and $r - f \geq 0$ additional independent columns. Using a phase I, one identifies $r$

independent rows in subset $R \subseteq \{1, \ldots, m\}$ and the vector subspace basis can take the form $\mathbf{\Lambda}_r = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{M} \end{bmatrix}$, where $\mathbf{M}$ is an $r \times (m-r)$ matrix, whereas $\mathbf{\Lambda}_r^\perp = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{m-r} \end{bmatrix}$. Let $\mathbf{V}_R$ be the vector subspace spanned by $\mathbf{\Lambda}_r$.

At the end of the day, the definition of compatibility can be enlarged to the spanning set of the chosen vector subspace basis $\mathbf{\Lambda}_r$. If the latter can be seen as a *poorer* decomposition which obviously includes $\mathbf{A}_F$, the added granularity provided by the superfluous columns yields a denser compatible set. Following Proposition 2, we introduce a second definition regarding compatibility.

**Definition 2** $\mathbf{a} \in \mathbb{R}^m$ *is* compatible with row-set R *if and only if it belongs to* $\mathbf{V}_R$*, the vector subspace of* $\mathbb{R}^m$ *spanned by* $\mathbf{\Lambda}_r$*.*

The danger of over-spanning $\mathbf{A}_F$ is that a compatible direction $\theta_\omega, \omega \in \Omega$, found by the pricing problem does not guarantee a strictly improving pivot. Indeed, any value $\bar{\mathbf{a}}^0_{i\omega} \neq 0, i \in R$, corresponding to $\bar{b}^0_i = 0, i \in R$, yields a zero-step size for column $\theta_\omega$, hence a degenerate pivot. Observe that the magnitude of this value is irrelevant, what really matters is the sign. In a very basic sense, the probability of making a degenerate pivot thus increases exponentially by one half for every extra rows, i.e., $(\frac{1}{2})^{r-f}$. A more refined analysis would include the density of matrix $\mathbf{A}$ and at this point falls outside the purpose of this paper.

# 4 Aiming for efficiency

This section serves the practical side of an implementation. With a better understanding of the pricing problem, Sect. 4.1 derives a *dynamic* Dantzig-Wolfe decomposition, that is, performed at every iteration. This is followed by three subsections which examine various ways to accelerate IPS in light of the considerations highlighted throughout the section.

## 4.1 A dynamic Dantzig-Wolfe decomposition

We now present a nice interpretation of IPS in terms of a decomposition scheme proposed by Metrane et al. (2010) for standard linear programs. Here is an adaptation for the bounded case.

Consider a Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960) of the previous so-called compact formulation (30) which has a block angular structure. The equality constraints in set $P$ together with the interval constraints $\mathbf{l}_F \leq \mathbf{x}_F \leq \mathbf{u}_F$ and upper bounds $\mathbf{y}_N \leq \mathbf{r}_N$ stay in the master problem structure. The equality constraints in row-set $Z$ and the non-negativity constraints $\mathbf{y}_N \geq \mathbf{0}$ form the subproblem domain:

$$\mathcal{SP} := \{\mathbf{y}_N \geq \mathbf{0} \mid \bar{\mathbf{A}}^0_{ZN}\mathbf{y}_N = \mathbf{0}\}.$$

It is well known (see Lübbecke and Desrosiers, 2005) that any vector $\mathbf{y}_N \in \mathcal{SP}$ can be reformulated as a convex combination of extreme points plus a non-negative combination of extreme rays of $\mathcal{SP}$. However, $\mathcal{SP}$ is a cone for which the only extreme point is the null vector $\mathbf{y}_N = \mathbf{0}$ at zero cost. Moreover, it has no contribution on the master problem constraints and as such it can be discarded from the reformulation. Vector $\mathbf{y}_N$ can thus be expressed as a non-negative combination of the extreme rays $\{\mathbf{y}^\omega_N\}_{\omega \in \Omega}$:

$$\mathbf{y}_N = \sum_{\omega \in \Omega} \mathbf{y}^\omega_N \theta_\omega,\ \theta_\omega \geq 0, \quad \forall \omega \in \Omega.$$

Substituting in the master problem structure, LP becomes:

$$\begin{array}{rrcrcll} z^\star := \mathbf{c}^\intercal_L\mathbf{x}^0_L + \mathbf{c}^\intercal_U\mathbf{x}^0_U + \min & \mathbf{c}^\intercal_F\mathbf{x}_F & + & \sum_{\omega \in \Omega}[\mathbf{d}^\intercal_N\mathbf{y}^\omega_N]\theta_\omega & & & \\ \text{s.t.} & \mathbf{A}_{PF}\mathbf{x}_F & + & \sum_{\omega \in \Omega}[\mathbf{A}^0_{PN}\mathbf{y}^\omega_N]\theta_\omega & = & \mathbf{b}^0_P, & [\boldsymbol{\psi}_P] \\ & & & \sum_{\omega \in \Omega}[\mathbf{y}^\omega_N]\theta_\omega & \leq & \mathbf{r}_N, & \\ & \mathbf{l}_F \leq \mathbf{x}_F \leq \mathbf{u}_F, & & \theta_\omega \geq 0,\ \forall \omega \in \Omega. & & & \end{array} \tag{32}$$

At any iteration of IPS, none of the $\boldsymbol{\theta}$-variables are yet generated and the inequality constraints in (32) are not binding. Therefore, the dual vector for these constraints is null and the reduced cost of variable $\theta_\omega, \omega \in \Omega$, is given by:

$$[\mathbf{d}_N^\intercal \mathbf{y}_N^\omega] - \boldsymbol{\psi}_P^\intercal [\mathbf{A}_{PN}^0 \mathbf{y}_N^\omega] = (\mathbf{d}_N^\intercal - \boldsymbol{\psi}_P^\intercal \mathbf{A}_{PN}^0)\mathbf{y}_N^\omega = \tilde{\mathbf{d}}_N^\intercal \mathbf{y}_N^\omega,$$

where $\tilde{\mathbf{d}}_N$ is the partial reduced cost vector already used in IPS, see formulation (12). Now, any negative reduced cost ray in $\mathcal{SP}$ results in the same subproblem minimum objective value, that is, $-\infty$. However, observe that for any non-zero solution in the cone defined by $\mathcal{SP}$, there exists a scaled one such that $\mathbf{1}^\intercal \mathbf{y}_N = 1$. Therefore, without loss of generality, the domain of the subproblem can be rewritten as

$$\mathcal{SP}_N := \{\mathbf{y}_N \geq \mathbf{0} \mid \bar{\mathbf{A}}_{ZN}^0 \mathbf{y}_N = \mathbf{0}, \quad \mathbf{1}^\intercal \mathbf{y}_N = 1\}.$$

Hence, an equivalent subproblem in this Dantzig-Wolfe decomposition, searching for a negative reduced cost column until optimality is reached, is exactly the one defined by the primal pricing problem (13) in IPS:

$$\min \ \tilde{\mathbf{d}}_N^\intercal \mathbf{y}_N \quad \text{s.t.} \quad \mathbf{y}_N \in \mathcal{SP}_N. \tag{33}$$

IPS corresponds to dynamically applying this Dantzig-Wolfe reformulation at every iteration, the row-decomposition being done according to the current solution vector $\mathbf{x}^0 = [\mathbf{x}_F^0, \mathbf{x}_L^0, \mathbf{x}_U^0]$. This interpretation of IPS can result in very flexible resolution strategies. Among these is the natural idea of generating several columns during one iteration of the pricing step. There is also that of working directly with the *surrogate $\boldsymbol{\theta}$-variables* issued from the chosen directions as opposed to decomposing the direction into its different components (the original $\mathbf{x}$-variables). The third and perhaps most important idea defers to the time consuming task of updating the row decomposition.

**Multiple improving directions.** With respect to finding multiple improving directions, it is a basic idea to harvest more information from the pricing problem than the one iteration. The price to pay is the possibility of making degenerate pivots on some of these directions. The one worthy mention is therefore the particular case of *independent improving directions*. In IPS, we rely on the strictly improving property of the algorithm to guarantee that the exchange mechanism *cycles* through the components of $\theta_\omega$ with a strictly positive step size, see (19). If two variables $\theta_{\omega_1}$ and $\theta_{\omega_2}$ can be identified from the pricing problem such that compatibility is obtained from orthogonal vectors of $\mathbf{V}$, $\bar{\mathbf{a}}_{\omega_1}^0$ and $\bar{\mathbf{a}}_{\omega_2}^0$ are *independent* from each other and can be added to the current solution in any order both yielding a predictable improvement.

**Surrogate variable vs. component wise.** Each new column $\theta_\omega$ can be split into its components, the original $\mathbf{x}$-variables. These can be sent to fill in the compact formulation (30) together with the constraints that are active in the subproblem solution. Similar techniques to solve large scale linear multi-commodity flow problems were previously used by Löbel (1998) and Mamer and McBride (2000), whereas Valério de Carvalho (1999, 2002) proposed a network-based compact formulation of the cutting stock problem where the classical knapsack subproblem is solved as a shortest path problem. In all these applications, the compact formulation is written in terms of arc flow variables. The subproblems generate *paths* with negative reduced costs but the *arcs* of these paths are iteratively added to the compact formulation. This process allows the implicit combination of arcs into paths without having to generate these. Sadykov and Vanderbeck (2013) describes this in generality.

Computational experiments conducted with a hybrid algorithm starting with the classical generated columns for the restricted master problem and ending with their components for the compact formulation show improving average factors of 3.32 and 13.16 compared to the primal simplex of `CPLEX` on the previously mentioned *simultaneous vehicle and crew scheduling problems in urban mass transit systems* and *aircraft fleet assignment* problems (Metrane et al., 2010).

## 4.2 Vector subspace basis update

From the Dantzig-Wolfe mindset, it becomes clear that entering an improving direction $\theta_w$ in the master problem (32) does not necessarily warrant an update of the vector subspace basis. Once again, the row decomposition is only the fruit of a linear transformation $\mathbf{T}^{-1}$ at a given iteration.

As long as the chosen vector subspace basis spans the columns associated with the free variables in the current solution, it is not necessary to modify the subspace basis. We argue that using surrogate variables allows to reuse the previous subspace basis because its very definition states that it is spanned by said basis. Unfortunately, when more than one former free variable becomes degenerate, the old subspace basis now generates basic degenerate variables. It is therefore possible to maintain the old subspace basis but it implies the use of the more general form $\mathbf{\Lambda}_r$. In accordance with Sect. 3.3, we will state that an update is in order when the actual number of free variables $f$ is relatively different from $r$, the row-size of the master problem. Let us look at the formulation.

By Definition 2, columns of $\mathbf{A}_F$ are compatible with $\mathbf{V}_R$. Denote by $\mathbf{A}^0_C, C \subseteq N$, the columns of $\mathbf{A}^0_N$ compatible with $\mathbf{V}_R$. Any of these can easily be identified in $O(m)$ time using the Positive Edge rule, see Sect. 7. Let $\mathbf{A}_I$ be the incompatible columns, $I := N \setminus C$. Using $\mathbf{T} := [\mathbf{\Lambda}_r, \mathbf{\Lambda}^\perp_r]$ and transformation $\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ -\mathbf{M} & \mathbf{I}_{m-r} \end{bmatrix}$ on formulation (3), we have $\bar{\mathbf{a}}^0_{Zj} = \bar{\mathbf{b}}^0_Z = \mathbf{0}, \forall j \in F \cup C$. Let $\bar{\mathbf{A}}^0_{ZI} := \mathbf{A}^0_{ZI} - \mathbf{M}\mathbf{A}^0_{RI}$. LP becomes

$$
\begin{array}{rlllll}
z^\star := \mathbf{c}_L^\intercal \mathbf{x}^0_L + \mathbf{c}_U^\intercal \mathbf{x}^0_U + \min & \mathbf{c}_F^\intercal \mathbf{x}_F & + \mathbf{d}_C^\intercal \mathbf{y}_C & + \mathbf{d}_I^\intercal \mathbf{y}_I & & \\
\text{s.t.} & \mathbf{A}_{RF}\mathbf{x}_F & + \mathbf{A}^0_{RC}\mathbf{y}_C & + \mathbf{A}^0_{RI}\mathbf{y}_I & = \mathbf{b}^0_R, & [\boldsymbol{\psi}_R] \\
& & & \bar{\mathbf{A}}^0_{ZI}\mathbf{y}_I & = \mathbf{0}, & [\boldsymbol{\psi}_Z] \\
& \multicolumn{5}{l}{\mathbf{l}_F \le \mathbf{x}_F \le \mathbf{u}_F, \quad \mathbf{0} \le \mathbf{y}_C \le \mathbf{r}_C, \quad \mathbf{0} \le \mathbf{y}_I \le \mathbf{r}_I.}
\end{array} \tag{34}
$$

Restricting the above formulation to column-index set $F \cup C$ (the set of compatible variables) yields a *row-reduced problem* defined on row-set $R$, much easier to solve than (1) as it involves fewer variables and, more importantly, fewer constraints. As such, it is less subject to degeneracy. However, its optimal solution does not necessarily solve LP. To have access to valid combinations of incompatible columns in $\mathbf{A}^0_I$, once again a Dantzig-Wolfe decomposition can be used, keeping in the pricing subproblem the equalities in $Z$, the non-negativity requirements $\mathbf{y}_I \ge \mathbf{0}$, and a scaling constraint on $\mathbf{y}_I$:

$$
\min \ \tilde{\mathbf{d}}_I^\intercal \mathbf{y}_I \quad \text{s.t.} \quad \mathbf{y}_I \in \mathcal{SP}_I := \left\{ \mathbf{y}_I \ge \mathbf{0} \mid \bar{\mathbf{A}}^0_{ZI}\mathbf{y}_I = \mathbf{0}, \ \mathbf{1}^\intercal \mathbf{y}_I = 1 \right\}. \tag{35}
$$

As previously derived in Sect. 4.1, the substitution of the extreme rays generated from $\mathcal{SP}_I$, $\mathbf{y}^\omega_I, \omega \in \Omega$, into the master problem gives

$$
\begin{array}{rlllll}
z^\star := \mathbf{c}_L^\intercal \mathbf{x}^0_L + \mathbf{c}_U^\intercal \mathbf{x}^0_U & & & & & \\
+ \min & \mathbf{c}_F^\intercal \mathbf{x}_F & + \mathbf{d}_C^\intercal \mathbf{y}_C & + \sum_{\omega\in\Omega} [\mathbf{d}_I^\intercal \mathbf{y}^\omega_I]\theta_\omega & & \\
\text{s.t.} & \mathbf{A}_{RF}\mathbf{x}_F & + \mathbf{A}^0_{RC}\mathbf{y}_C & + \sum_{\omega\in\Omega} [\mathbf{A}^0_{RI}\mathbf{y}^\omega_I]\theta_\omega & = \mathbf{b}^0_R, & [\boldsymbol{\psi}_R] \\
& & & \sum_{\omega\in\Omega} [\mathbf{y}^\omega_I]\theta_\omega & \le \mathbf{r}_N, & \\
& \multicolumn{5}{l}{\mathbf{l}_F \le \mathbf{x}_F \le \mathbf{u}_F, \quad \mathbf{0} \le \mathbf{y}_C \le \mathbf{r}_C, \quad \theta_\omega \ge 0, \forall \omega \in \Omega,}
\end{array} \tag{36}
$$

and similarly to (9)–(10) or (31), dual vector $\boldsymbol{\pi}$ can be retrieved using the expression $\boldsymbol{\pi}^\intercal = \boldsymbol{\psi}^\intercal \mathbf{T}^{-1}$:

$$
[\boldsymbol{\pi}_R^\intercal, \boldsymbol{\pi}_Z^\intercal] = \left[ \ \boldsymbol{\psi}_R^\intercal - \boldsymbol{\psi}_Z^\intercal \mathbf{M}, \ \boldsymbol{\psi}_Z^\intercal \ \right], \tag{37}
$$

where $\boldsymbol{\psi}_Z$ is the dual vector of constraint set $\bar{\mathbf{A}}^0_{ZI}\mathbf{y}_I = \mathbf{0}$ in pricing problem (35). The column generation process stops when no negative reduced cost columns exist in $\mathcal{SP}_I$.

## 4.3 Vector subspace flexibility

Since the vector subspace is defined with respect to matrix $\mathbf{A}_F$ of free variables, this section shows that it is even possible to play with the set of free variables as we see fit. The first trick considers a particular type of upper bounds while the other cheats the free status with algebraic manipulations.

**Implicit upper bounds.** Taking upper bounds into account is an obligatory path to guarantee strictly improving directions. However, some applications have a structure that *implicitly* bounds some variables by the sheer force of the technological constraints. For instance, the assignment and the set partitioning models have such a feature. As a matter of fact, all variables in both of these problems are bounded above by 1, yet the explicit bound constraint needs not be added to the formulation. That is to say that a variable $x$ features an *implicit* upper bound $\upsilon$ if $x > \upsilon$ is infeasible regardless of the values of the remaining variables.

IPS can therefore be applied in two different manners with respect to the way the upper bounds are taken into account. The main difference echoes the vector subspace $\mathbf{V}$ and thus the set of compatible variables, see Proposition 2. Consider vector subspaces spanned by $\mathbf{A}_F$ which contains or not implicit bounded variables. It is obvious that the direction set of their respective pricing problem is the same for either subspaces. The distinction lies in the compatibility set and different $\mathbf{A}_F$ modify the relative row-sizes of the master $(f)$ and the subproblem $(m-f+1)$. The added granularity provided by the additional vectors in the first case creates a denser linear span and thus allows more variable compatibility.

**Coerced degeneracy.** Another highly important concept is that of *coerced* degeneracy. This is used in the capacitated minimum cost network flow problem which can artificially render any current free variable into two degenerate ones on the residual network, see Ahuja et al. (1993). Indeed, a variable $x_{ij}$ taking a value $\ell_{ij} < x^0_{ij} < u_{ij}$ of the original network formulation can be replaced by two variables representing upwards $(0 \leq y_{ij} \leq u_{ij} - x^0_{ij})$ and downwards $(0 \leq y_{ji} \leq x^0_{ij} - \ell_{ij})$ possible flow variations. Current value of these $\mathbf{y}$-variables is null and again this can modify the relative row-sizes of the master and the subproblem. On either count, the choice of the vector subspace results in a degenerate free pricing step.

### 4.4 Partial pricing

In this subsection, we discuss possible partial pricing choices to accelerate the resolution process without compromising optimality.

**Cost and residual capacity bias.** Partial pricing strategies become appealing in diversified aspects. For example, one can use various subsets of compatible and incompatible variables: only those with negative partial reduced costs $\tilde{d}_j, j \in I$ (ensuring a negative reduced cost convex combination); only those with a relatively large range $r_j, j \in F \cup C \cup I$ (hoping for a large step size $\rho$); or only those sharing both properties. This can even be strengthened by temporarily enlarging row-set $Z$ (hence restricting row-set $R$) to avoid pivoting in (19) on small right-hand side values $\bar{b}^0_i - l_i$ or $u_i - \bar{b}^0_i, i \in R$.

**Rank-incompatibility bias.** Another possibility is to define the pricing step against *rank-incompatibility*. This means that the incompatible variables are attributed a rank according to the degree of incompatibility they display. The pricing problem sequentially uses lower rank incompatible variables. Intuitively, the point is not to stray too far from compatibility and thus not perturb too much the current compatibility definition. This concept was first seen in Dynamic Constraint Aggregation in the paper of Elhallaoui et al. (2010).

## 5 Designing around compatibility

As supported by the subspace and the subspace basis flexibility, the compatibility notion is indeed quite flexible. In fact, the specialized algorithms to solve particular linear programs are designed around and for compatibility.

### 5.1 Network flow

In the context of the capacitated minimum cost flow problem, one refers to a solution $\mathbf{x}$ as a cycle free solution if the network contains no cycle composed only of free arcs, see Ahuja et al. (1993). Any such solution can be represented as a collection of free arcs (the non-degenerate arcs forming a forest) and all other arcs at their lower or upper bounds (the degenerate ones). The column-vectors of the free arcs form $\mathbf{A}_F$, see Fig. 1.

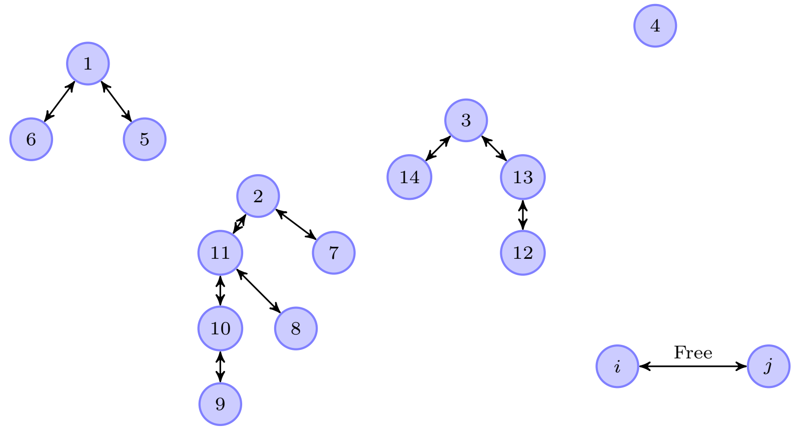


Figure 1: Forest of free arcs in $\mathbf{A}_F$ on a residual network

According to Proposition 2 and the flow conservation equations, a degenerate arc is compatible if and only if it can be written in terms of the unique subset of free arcs forming a cycle with it (Desrosiers et al., 2013b). Therefore, a compatible arc simply links two nodes belonging to the same tree of the forest. By opposition, an incompatible arc links two nodes of two different trees.

This characterization allows us to better understand the mechanism of improving cycles in networks. A feasible solution is optimal if and only if there is no negative cost directed cycle on the residual network. Two types of cycles can be the result of the pricing problem: a cycle containing a single compatible arc together with some free arcs of the same tree, or a cycle containing at least two incompatible arcs together with possibly some free arcs from different trees of the forest.

In Fig. 2, dotted arc $(8, 9)$ is compatible and forms a directed cycle with free arcs $(9, 10)$, $(10, 11)$ and $(11, 8)$. Indeed, associated column vectors in rows 8 through 11 are such that

$$\begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} = - \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} - \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

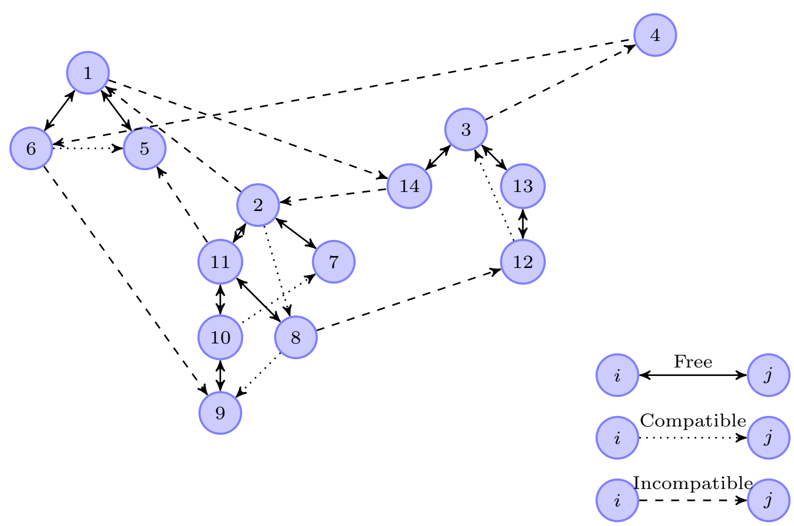


Figure 2: Compatibility characterization of degenerate arcs on a residual network

Dashed arc $(6, 9)$ links two different trees and is therefore incompatible. This is also the case for some other arcs, e.g., $(8, 12)$, $(3, 4)$ and $(4, 6)$. The reader may verify that the sum of the associated four column vectors is compatible as it can be written as the negated sum of the column vectors associated with free arcs $(9, 10)$, $(10, 11)$, $(11, 8)$ and $(12, 13)$, $(13, 3)$. Indeed, these nine arcs form a directed cycle in the residual network.

IPS only makes non-degenerate pivots while solving linear programs. Hence, this primal simplex-type algorithm takes a finite number of iterations on integral data network flow problems. Desrosiers et al. (2013a) show that IPS is strongly polynomial for binary network problems, e.g., assignment, single shortest path, unit capacity maximum flow. With a slight modification, it becomes strongly polynomial for solving the capacitated minimum cost network flow problem. The proposed contraction-expansion IPS-based algorithm is similar to the minimum mean cycle-canceling algorithm, see Goldberg and Tarjan (1989); Radzik and Goldberg (1994). On a network comprising $n$ nodes and $m$ arcs, it performs $O(mn)$ iterations and runs in $O(m^3n^2)$ time for arbitrary real-valued arc costs.

## 5.2 Set partitioning

The set partitioning problem (SPP) can be formulated as the binary linear program

$$\min \quad \mathbf{c}^\intercal \mathbf{x} \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{1}, \quad \mathbf{x} \in \mathbb{B}^n, \tag{38}$$

where $\mathbf{A} \in \mathbb{B}^m \times \mathbb{B}^n$. This formulation can be seen as a generic model encountered in various applications, namely, in vehicle routing and crew scheduling, and many more. In such applications, each set partitioning constraint can be associated with a task $i \in \{1, \ldots, m\}$ that must be *covered* exactly once. Such a formulation arises naturally from applying a Dantzig-Wolfe reformulation (Dantzig and Wolfe, 1960) to a multi-commodity flow model in which each vehicle or crew member is represented by a separate commodity, see Desrosiers et al. (1995) and Desaulniers et al. (1998). SPP probably has the strongest LP structure which falls short from the integrality property provided by the unimodularity characterization of network flow problems. This means that a basic solution to the linear relaxation of (38) may be either binary or fractional.

In order to express the fundamental exchange mechanism of set partitioning solutions, we assume that current vector $\mathbf{x}_F$ is binary. Figure 3 should help demystify the interpretation of compatibility on SPP.

$$\begin{array}{ccc|c|cc|ccc}
 & \mathbf{A}_F & & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 \\
\hline
1 & & & 1 & 1 & & & & \\
1 & & & 1 & & 1 & & & \\
 & 1 & & & 1 & & 1 & & 1 \\
 & 1 & & & & 1 & 1 & 1 & \\
 & 1 & & & & 1 & & 1 & 1 \\
 & & 1 & 1 & & & & 1 & 1
\end{array}$$

Figure 3: Compatibility characterization for set partitioning binary solution $\mathbf{x}_F$

In the left-hand side, we find the binary input solution defined by the three independent columns of $\mathbf{\Lambda}_f = \mathbf{A}_F$. According to Proposition 2, the next column identified by $x_4$ is compatible with the given partition as it covers exactly the first and third groups. The third set shows the two incompatible columns $x_5$ and $x_6$. None can be generated by the columns of $\mathbf{A}_F$. However, their addition is compatible with the given partition as it covers the first and second groups of rows. Finally, the right-hand side set exhibits three incompatible columns, $x_7$, $x_8$ and $x_9$: their combination with equal weights of $1/2$ is compatible as it covers the second and third groups of the row partition. Note that this combination breaks the integrality of the next solution.

The compatible directions are readily available as the spanning set of $\mathbf{A}_F$ as per Proposition 2: a binary column is compatible if and only if it covers some of the groups. Therefore, the interpretation of compatibility can be seen as a question of coverage. When the direction is a combination of incompatible variables, the exchange mechanism removes elements from some groups to insert them back in other groups.

When the input solution is fractional, the mathematical definition of compatibility (Definition 1 or Proposition 2) still holds but the interpretation looses practical meaning. In order to sidestep this unfortunate loss, we can fallback on $\mathbf{\Lambda}_r$ (Definition 2) and adjust the subspace basis interpretation with respect to an aggregation/grouping scheme. The idea is to assume that certain tasks are done together and it is the cornerstone of Dynamic Constraint Aggregation (DCA) as described in the following section. Historically speaking, DCA is a self-standing algorithm devised for set partitioning model which provides an easy way to define a specialized vector subspace that *often* shares the properties of the one designed for IPS. The next section discusses the differences and similarities that arise between the two methods. We insist that it is in retrospective that the ties between DCA and IPS have been better understood.

# 6 Dynamic Constraint Aggregation

It is the first time DCA and IPS are studied in parallel. If they share several similarities, we hope to dissolve the confusion that arises between the two theories by highlighting their differences. IPS relies on the linear algebra framework to ascertain its faith and is therefore constructive by nature. It turns out that DCA is also born from a constructive design. This design is however limited by the embryonic intuition of a *reduced basis*. Let it be said that DCA is an intuitive method precursor to IPS.

In a nut shell, the differences spring forth from the choice of the vector subspace to represent the current solution. Recall that IPS defines the vector subspace basis $\mathbf{\Lambda}_f$ as $\mathbf{A}_F$ (or an equivalent matrix) from the overrepresented vertex. DCA disregards this choice but constructs $\mathbf{\Lambda}_r, r \geq f$, large enough to span $\mathbf{A}_F$. We will see how and why it performs well.

DCA is devised solely for the set partitioning problem. It capitalizes on the compatibility interpretation and characterization of set partitioning optimal solutions. A binary solution to (38) is usually highly degenerate. Indeed, in typical applications, a group (variable) covers several tasks, say 10%, which implies that the number of variables assuming a positive value in the basis is of the same order.

## 6.1 $\mathbf{\Lambda}_r$ derived from the identical rows of $\mathbf{A}_F$

The mathematical idea behind DCA is similar to the first step of a LU-decomposition. Some of the rows which can easily be identified as null entries after elimination are actually identical rows. Restricting the decomposition to these manipulations induces a partition of the rows. Such a strategy might propose a set of constraints where some rows are redundant because linear independence is not thoroughly verified. The size of the partition is expressed as the number of *different rows* $r \geq f$ in $\mathbf{A}_F$.

$$\begin{array}{c}\mathbf{A}_F\\ \begin{pmatrix}\mathbf{r}_1\\ \mathbf{r}_1\\ \mathbf{r}_2\\ \mathbf{r}_3\\ \mathbf{r}_3\\ \mathbf{r}_3\end{pmatrix}\end{array} \mapsto \begin{pmatrix}1 & & \\ 1 & & \\ & 1 & \\ & & 1\\ & & 1\\ & & 1\end{pmatrix} \equiv \left(\begin{array}{c}\mathbf{r}_1\\ \mathbf{r}_2\\ \mathbf{r}_3\\ \hline \mathbf{r}_1\\ \mathbf{r}_3\\ \mathbf{r}_3\end{array}\right) \mapsto \begin{array}{c}\mathbf{\Lambda}_r\\ \left(\begin{array}{ccc}1 & & \\ & 1 & \\ & & 1\\ \hline 1 & & \\ & & 1\\ & & 1\end{array}\right)\end{array}$$

The vector subspace basis $\mathbf{\Lambda}_r$ may over-span $\mathbf{A}_F$ if $r > f$. In other words, when $r = f$ we get from Proposition 2 that the decomposition is minimal and exactly corresponds to a generic basis of IPS. When $r > f$, $\mathbf{\Lambda}_r$ may lead to degenerate pivots although hopefully less than with the primal simplex algorithm.

## 6.2 DCA on set partitioning models

In SPP, wishful thinking unites with practicality. Indeed, the binary structure of the technological constraints is highly prejudicial to identical appearances in the subspace represented by positive variables. In fact, when solution is binary the partition always corresponds to the working basis of IPS. That is to say that when we aim to maintain integrality in the resolution process, it makes it hard not to endorse DCA's strategy. Assume the linear relaxation of the set partitioning formulation (38) is written in standard form, that is,

$$z^\star = \min \quad \mathbf{c}^\intercal \mathbf{x} \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{1}, \quad \mathbf{x} \geq \mathbf{0}. \tag{39}$$

We present three examples of DCA. The first assumes the current solution is binary while the second and third consider a fractional input for which the partition is the same as the IPS decomposition for the former and different for the latter.

If $\mathbf{x}_F$ is binary, the corresponding columns of $\mathbf{A}_F$ are disjoint and induce a partition of the row-set into $f$ groups. From $\mathbf{A}_F$, it is easy to construct a reduced working basis: take a single row from each group and therefore, upon a reordering of the columns and rows of $\mathbf{A}$, matrix $\mathbf{A}_{PF}$ is $\mathbf{I}_f$. This is illustrated with the following integer solution $(x_1, x_2, x_3) = (1, 1, 1)$:

$$\begin{array}{c}\mathbf{A}_F\\ \left(\begin{array}{ccc} 1 & & \\ 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & & 1 \\ & & 1 \end{array}\right)\end{array} \mapsto \begin{array}{c}\mathbf{\Lambda}_f\\ \left(\begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \\ \hline 1 & & \\ 1 & & \\ & 1 & \\ & & 1 \end{array}\right)\end{array}.$$

If the solution $\mathbf{x}_F > \mathbf{0}$ is fractional, the partition of the row-set is derived from the $f$ groups of *identical* rows of $\mathbf{A}_F$, the same rule as for a binary solution. To construct a working basis, take the first row from each group to form $\mathbf{A}_{PF}$ while the $m - f$ rows of $\mathbf{A}_{ZF}$ are copies of a selection of the $f$ independent rows of $\mathbf{A}_{PF}$. However, right multiplying $\mathbf{A}_F$ by $\mathbf{A}_{PF}^{-1}$ provides the alternative vector subspace basis $\mathbf{A}_F\mathbf{A}_{PF}^{-1} = \begin{bmatrix} \mathbf{I}_f \\ \mathbf{A}_{ZF}\mathbf{A}_{PF}^{-1} \end{bmatrix}$ to the vector subspace spanned by $\mathbf{A}_F$. This alternative vector subspace basis is similar to the one obtained from a binary solution. This is illustrated with the following 3-variable fractional solution $(x_1, x_2, x_3) = (0.5, 0.5, 0.5)$:

$$\begin{array}{c}\mathbf{A}_F\\ \left(\begin{array}{ccc} & 1 & 1 \\ & 1 & 1 \\ & 1 & 1 \\ 1 & & 1 \\ 1 & & 1 \\ 1 & 1 & \\ 1 & 1 & \\ 1 & 1 & \end{array}\right)\end{array} \mapsto \left(\begin{array}{ccc} & 1 & 1 \\ 1 & & 1 \\ 1 & 1 & \\ \hline & 1 & 1 \\ & 1 & 1 \\ 1 & & 1 \\ 1 & 1 & \\ 1 & 1 & \end{array}\right) \equiv \begin{array}{c}\mathbf{\Lambda}_{r=f}\\ \left(\begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \\ \hline 1 & & \\ 1 & & \\ & 1 & \\ & & 1 \\ & & 1 \end{array}\right)\end{array}.$$

The final example involves four fractional variables in $\mathbf{A}_F$ with $(x_1, x_2, x_3, x_4) = (0.5, 0.5, 0.5, 0.5)$ but the vector subspace basis $\mathbf{\Lambda}_r$ has dimension $r > f$ in the partition induced by the identical rows:

$$\begin{array}{c}\mathbf{A}_F\\ \left(\begin{array}{cccc} 1 & 1 & & \\ & 1 & 1 & \\ & & 1 & 1 \\ 1 & & 1 & \\ 1 & & & 1 \\ 1 & 1 & & \\ & 1 & 1 & \\ 1 & & & 1 \end{array}\right)\end{array} \mapsto \begin{array}{c}\mathbf{\Lambda}_{r>f}\\ \left(\begin{array}{ccccc} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \\ \hline 1 & & & & \\ & 1 & & & \\ & & & & 1 \end{array}\right)\end{array}.$$

In the context of routing and scheduling, *compatibility* thus means generating itineraries or schedules which precisely respect the current aggregation/grouping into multi-task activities. Of course, this is in perfect harmony with Definition 2. ¿From the above discussion, we see that the vector subspace basis $\mathbf{\Lambda}_r$ is derived from the solution of the linear relaxation formulation (39). However, the process can be initialized from any partition $\mathbf{\Lambda}_r$ of the rows: this can be done in a heuristic manner, even infeasible. This simply results in a linear transformation $\mathbf{T}^{-1}$ applied on the system of equality constraints, updated when needed.

### 6.3 Resolution process

Assume that the linear relaxation of the set partitioning formulation (38) is written in standard form (39). Derived from vector basis $\mathbf{\Lambda}_r$, consider the row-partition $\{R_1, \ldots, R_r\}$, where $f \leq r \leq m$. A row-reduced or aggregated problem, evocative of the restricted master problem of a Dantzig-Wolfe decomposition, can be defined by taking the first row of each group to form row-set $R$. The aggregation comes from the partition while the sense of restricted is attributed to the concept of compatibility. When the row-reduced problem in formulation (34) is solved to optimality using the variables in $F \cup C$,

$$\begin{array}{llllll} z^\star := \min & \mathbf{c}_F^\intercal \mathbf{x}_F & + & \mathbf{c}_C^\intercal \mathbf{y}_C & & \\ \quad \text{s.t.} & \mathbf{A}_{RF}\mathbf{x}_F & + & \mathbf{A}_{RC}\mathbf{y}_C & = \mathbf{1}, & [\boldsymbol{\psi}_R] \\ & \mathbf{x}_F \geq \mathbf{0}, & & \mathbf{y}_C \geq \mathbf{0}, & & \end{array} \tag{40}$$

it remains to be seen if the vector of incompatible variables $\mathbf{y}_I$ agrees as well. While IPS provides a pricing problem (35) to answer this question, DCA did not have such a feature when it was originally designed.

Let's go back to the basics. From now on assume $F$ represents the index set of the current positive (or free) variables in the optimal solution to formulation (40), where $f \leq r$, and $\boldsymbol{\psi}_R$ is an optimal dual vector. Variables $\mathbf{x}_F$ and $\mathbf{y}_C$ being compatible, we already have $\bar{\mathbf{c}}_F = \mathbf{c}_F - \boldsymbol{\psi}_R^\intercal \mathbf{A}_{RF} = \mathbf{0}$ and $\bar{\mathbf{c}}_C = \mathbf{c}_C - \boldsymbol{\psi}_R^\intercal \mathbf{A}_{RC} \geq \mathbf{0}$, and current solution is optimal if

$$\bar{\mathbf{c}}_I = \mathbf{c}_I - \boldsymbol{\psi}_R^\intercal \mathbf{A}_{RI} - \boldsymbol{\psi}_Z^\intercal \bar{\mathbf{A}}_{ZI} \geq \mathbf{0}.$$

However, since the neglected constraints in row-set $Z$ have no dual information on $\boldsymbol{\psi}_Z$, it seems like we have a bit of freedom for the computation of said reduced costs $\bar{\mathbf{c}}_I$. Well, it might be a tad rash to speak of freedom; the current optimal state is indeed conditional upon the chosen partition. Equivalently, $\bar{\mathbf{c}}_I = \mathbf{c}_I - \boldsymbol{\pi}^\intercal \mathbf{A}_I$, and dual vector $\boldsymbol{\psi}_R$ must thus be adapted to $\boldsymbol{\pi}$. The answer appears in (37), $[\boldsymbol{\pi}_R^\intercal, \boldsymbol{\pi}_Z^\intercal] = [\boldsymbol{\psi}_R^\intercal - \boldsymbol{\psi}_Z^\intercal \mathbf{M}, \boldsymbol{\psi}_Z^\intercal]$, which leads us to

$$\boldsymbol{\pi}_R^\intercal + \boldsymbol{\pi}_Z^\intercal \mathbf{M} = \boldsymbol{\psi}_R^\intercal. \tag{41}$$

As every column of the binary matrix $\mathbf{M}$ in a set partitioning problem identifies the remaining rows of a group, it means that $\psi_i, i \in R$, an aggregated dual variable, must basically be distributed across the rows of its group in set $R_i$, that is, $\sum_{\ell \in R_i} \pi_\ell = \psi_i$. Note that $\forall i \in R$, no matter how the $\boldsymbol{\psi}_i$ are distributed over their respective groups, $\bar{\mathbf{c}}_F = \mathbf{0}$ and $\bar{\mathbf{c}}_C \geq \mathbf{0}$ remain satisfied. Therefore, consider the following set of constraints:

$$\boldsymbol{\pi}^\intercal \mathbf{a}_j \leq c_j, \quad \forall j \in I \tag{42}$$

$$\sum_{\ell \in R_i} \pi_\ell = \psi_i, \quad \forall i \in R. \tag{43}$$

Notice that system (42)–(43) is about feasibility. On the one hand, it is indeed feasible which means that the existence of acceptable dual variables certifies the optimal status of current solution. On the other hand, it is infeasible: some constraints from (42) are removed until one retrieves a vector $\boldsymbol{\pi}$. Then a small selection of would-be negative reduced costs incompatible variables is presumptuously added to current $\mathbf{A}_F$, say columns in $\mathbf{A}_{I'}, I' \subset I$, such that a new partition is induced by the identical rows in $[\mathbf{A}_F, \mathbf{A}_{I'}]$. This yields a new vector subspace basis $\mathbf{\Lambda}_r, r > f$, at which point the algorithm proceeds with a new iteration.

The exercise of distributing the dual multipliers is called *dual variable disaggregation* in DCA. Since the expectation of an optimal solution to (42)–(43) can be put on hold, the system can be preemptively constructed in such a way that the algorithm expects a new partition. In particular, Elhallaoui et al. (2010) use a low-rank incompatibility strategy. In this respect, the disaggregation is heuristic by design. The pricing step established in IPS is however now available for DCA.

### 6.4 Computational results

The idea of compatibility steered the research and the implementation of DCA and its variants. For problems with about 2,000 constraints and 25,000 variables, with on average 40% to 50% positive variables, the combination of these ideas within the GENCOL software system allows a reduction of solution times by factors of 50 to 100 (4.86 DCA $\times$ 4.38 BDCA $\times$ 4.53 MPDCA). The improvement factors are compounded as discussed in the listed papers. The last two acronyms respectively refer to Bi-DCA where tasks are also aggregated within the constrained shortest path subproblem solution process and Multi-phase DCA where a low-rank incompatibility strategy is used for generating negative reduced cost columns.

To overcome degeneracy encountered during the solution of the restricted master problems in branch-and-price, Benchimol et al. (2012) propose a *stabilized* DCA, called SDCA, that incorporates the above mentioned DCA into the dual variable stabilization (DVS) method of Oukil et al. (2007). The rationale behind this combination is that DCA reduces the primal space whereas DVS acts in the dual space. Thus, combining both allows to fight degeneracy from primal and dual perspectives simultaneously. This method is again designed for solving the linear relaxation of set partitioning type models only. The computational results obtained on randomly generated instances of the multi-depot vehicle scheduling problem show that the performance of SDCA is not affected by the level of degeneracy and that it can reduce the average computational time of the master problem by a factor of up to 7 with respect to DVS. While this is not a direct comparison with DCA, the reduction factor would be even greater. Indeed, the many instances solved by DVS could not be solved by DCA alone.

## 7 Positive Edge

As in any simplex type algorithm, various pricing strategies can be used in IPS. Researchers suggested to first consider non-basic variables that are *compatible with row-set* $P$, or *compatible with row-set* $R$ in the general case, i.e., those in set $C := \{j \in N \mid \bar{\mathbf{a}}^0_{Zj} = \mathbf{0}\}$, followed by the more involved pricing step on the incompatible ones in set $I := \{j \in N \mid \bar{\mathbf{a}}^0_{Zj} \neq \mathbf{0}\}$. However, the identification of variables compatible with row-set $R$ using Definition 2 requires the computation of the transformed matrix $\bar{\mathbf{A}}^0_{ZN} = \mathbf{T}^{-1}_Z \mathbf{A}^0_{ZN}$, where $\mathbf{T}^{-1}_Z := \begin{bmatrix} -\mathbf{M} & \mathbf{I}_{m-r} \end{bmatrix}$. For large-scale problems, this can be time consuming.

### 7.1 Observations

Because the sign is irrelevant, $\bar{\mathbf{a}}^0_{Zj} = \mathbf{0}$ is equivalent to $\bar{\mathbf{a}}_{Zj} = \mathbf{0}$, $\forall j \in N$. The *Positive Edge* rule (Raymond et al., 2010a) allows to determine whether a variable $y_j, j \in N$, is compatible or not without explicitly computing vector $\bar{\mathbf{a}}_{Zj} = \mathbf{T}^{-1}_Z \mathbf{a}_j$. The PE rule is based on the following observations. Let $\mathbf{v} \in \mathbb{R}^{m-r}_0$ be a random vector for which all components are different from zero. If $\mathbf{a}_j$ is compatible, then $\bar{\mathbf{a}}_{Zj} = \mathbf{0}$ and hence $\mathbf{v}^\intercal \bar{\mathbf{a}}_{Zj} = \mathbf{0}$. Otherwise $\bar{\mathbf{a}}_{Zj} \neq 0$ and

$$\mathbf{v}^\intercal \bar{\mathbf{a}}_{Zj} = 0 \quad \text{if and only if} \quad \mathbf{v} \perp \bar{\mathbf{a}}_{Zj}, \tag{44}$$

that is, if and only if $\mathbf{v}$ and $\bar{\mathbf{a}}_{Zj}$ are orthogonal, which has probability zero for a continuous random vector $\mathbf{v} \in \mathbb{R}^{m-r}_0$. Define $\mathbf{w}^\intercal := \mathbf{v}^\intercal \mathbf{T}^{-1}_Z$. Then, for any variable $y_j, j \in N$,

$$\mathbf{v}^\intercal \bar{\mathbf{a}}_{Zj} = \mathbf{v}^\intercal \mathbf{T}^{-1}_Z \mathbf{a}_j = \mathbf{w}^\intercal \mathbf{a}_j, \tag{45}$$

and one can use the PE rule $\mathbf{w}^\intercal \mathbf{a}_j = 0$ for a compatibility-test using original vector $\mathbf{a}_j$. Expression (45) is similar to $\mathbf{c}^\top_B \bar{\mathbf{a}}_j = \boldsymbol{\pi}^\top \mathbf{a}_j$ in the computation of the reduced cost of variable $x_j$, where $\boldsymbol{\pi}^\top := \mathbf{c}^\top_B \mathbf{B}^{-1}$ is the vector of dual variables associated with constraint set $\mathbf{Ax} = \mathbf{b}$.

Of course, the computer architecture obliges to consider the rational set. According to Raymond et al. (2010a), a random vector $\mathbf{v} \in \mathbb{Q}_0^{m-r}$, chosen on a computer with 32-bit words, is such that all components are independent and identically distributed: $v_i \sim SEM_{32}, i \in Z$.

**Definition 3** *A floating-point number $\mathcal{F}$ with distribution $SEM_{32}$ is a single precision number where the sign-bit $S$, the exponent field $E$, and the mantissa field $M$ are independent and follow the discrete uniform distributions $S \sim U[0,1]$, $E \sim U[64,191]$, and $M \sim U[0, 2^{23}-1]$.*

Discrete distribution $SEM_{32}$ is symmetric around a zero mean ($\mu_{\mathcal{F}} = 0$) with a huge dispersion, its standard deviation being $\sigma_{\mathcal{F}} > 2^{60}$ (Towhidi et al., 2012).

**Positive Edge rule.** Let $\mathbf{v} \in \mathbb{Q}_0^{m-r}$ be a random vector, $v_i \neq 0, \forall i \in Z$. Compute $\mathbf{w}^\intercal := \mathbf{v}^\intercal \mathbf{T}_Z^{-1}$. $\mathbf{a} \in \mathbb{R}^m$ is *considered compatible with vector space* $\mathbf{V}_R$ if $\mathbf{w}^\intercal \mathbf{a}_j = \mathbf{0}$.

Since the operation amounts to a dot product, this means that a compatible variable is recognized in $O(m)$ time. The pricing problem over the non-basic ones reduces to $\min_{j \in C} \tilde{d}_j$ while that over the incompatible ones is identical to (13) but with $\mathbf{y}_N$ replaced by $\mathbf{y}_I$ or identical to (24) in the case of a linear program in standard form with $\mathbf{x}_N$ replaced by $\mathbf{x}_I$. Notice that considering set $C$ is solely from a theoretical point of view, while in practice the compatibility-test is only performed for variables with negative reduced costs or a subset of them.

## 7.2 Computational results

Although closely related to IPS, this PE rule for the identification of the compatible variables and the selection of these as entering ones in the primal simplex algorithm has only been tested independently of the Improved Primal Simplex. The proof of concept is provided in Raymond et al. (2010a) with the use of two external procedures with CPLEX while a direct implementation within COIN-OR's CLP, where it has been combined with the Devex pricing criterion, is presented in Towhidi et al. (2012). The proposed implementation uses a two-dimensional rule: for a variable $x_j, j \in N$, the first dimension computes the reduced cost $\bar{c}_j = c_j - \boldsymbol{\pi}^\intercal \mathbf{a}_j$, whereas the second evaluates $\mathbf{w}^\intercal \mathbf{a}_j$. PE identifies $C_{\mathbf{w}} = \{j \in N | \mathbf{w}^\intercal \mathbf{a}_j = 0\}$ and $I_{\mathbf{w}} = \{j \in N | \mathbf{w}^\intercal \mathbf{a}_j \neq 0\}$. Let $\bar{c}_{j_\star}, j_\star \in C_{\mathbf{w}} \cup I_{\mathbf{w}}$, be the smallest reduced cost and $\bar{c}_{j_{\mathbf{w}}}$, $j_{\mathbf{w}} \in C_{\mathbf{w}}$, be the smallest one for a compatible variable. The current solution is optimal if $\bar{c}_{j_\star} \geq 0$. Compatible variables are preferred to enter the basis except if $\bar{c}_{j_\star}$ is much smaller than $\bar{c}_{j_{\mathbf{w}}}$. Given $0 \leq \alpha < 1$, the selection rule is: if $\bar{c}_{j_\star} < 0$ and $\bar{c}_{j_{\mathbf{w}}} < \alpha \bar{c}_{j_\star}$, then select $x_{j_{\mathbf{w}}}$ else $x_{j_\star}$.

Tested on 32 problems from the Mittelmann's library which contains instances with a wide range of degeneracy levels, computational results show that below a degeneracy level of 25%, PE is on average neutral while above this threshold, it reaches an average run-time speedup of 2.72, with a maximum at 4.23 on an instance with a 75% degeneracy level.

# 8 Conclusions

This paper presents a survey of three recent tools for dealing with primal degeneracy in linear programming. Historically, the Dynamic Constraint Aggregation method (DCA) appeared first. It is specifically designed for set partitioning models encountered in many routing and scheduling formulations solved by branch-and-price. The Improved Primal Simplex algorithm (IPS) extended the concept to linear programs in general. Both methods partition the set of constraints in two parts, at every iteration, based on the values taken by the basic variables. This can be seen as a dynamic application of the Dantzig-Wolfe decomposition principle.

More specifically in IPS, one part of the constraints appears in the pricing problem as a homogeneous linear system (together with non-negative variables) while the other part (together with the bound intervals on the variables) is used in the master problem to complete the exchange mechanism from one feasible solution to the next. The third tool, the Positive Edge rule (PE), determines in polynomial time column-vectors to price out that are compatible with the current row partition, that is, it finds those entering variables that lead to non-degenerate pivots, if any. Otherwise, a convex combination of incompatible ones is identified in the IPS pricing step which also leads to a non-degenerate pivot until optimality is reached in a finite number of iterations. Computational results reported from the literature show a large reduction on CPU times attributed to the reduction of degenerate pivots.

This paper also unifies IPS and DCA through a new interpretation in terms of the usage of two different vector subspace bases spanning the columns of the master problem. On the one hand, the vector subspace basis of IPS is made of the column-vectors associated with the non-degenerate (or free) variables. On the other hand, that in DCA is derived from a partition of the rows into groups, as the one observed in any integer solution. This vector subspace basis has the fundamental property that it spans the free variable vectors. Therefore, the dimension of the vector subspace basis in DCA may be sometimes larger rather than equal to the number of free variables and this is the reason why some degenerate pivots may occur. As such, while every iteration of IPS is non-degenerate, DCA may encounter some degenerate pivots. The role of PE turns out to be the identification in the original formulation of the vectors that belong to the vector subspace spanned by the selected vector subspace basis.

What does the future look like? Future researches certainly concern implementation issues. We are essentially working on accelerating strategies to benefit from the various properties of the methods. The reader can also think of several specializations of IPS to well structured problems such as network flows, multi-commodity flows, and linear relaxations of set partitioning, set covering and bin packing problems. We are also looking for an implementation of this methodology within column generation, its adaptation to the dual simplex algorithm and to convex optimization, and its impact on the right-hand side sensitivity analysis, indeed the interpretation of the dual variables in the context of optimal degenerate solutions. Finally, the design of an efficient Integral Simplex algorithm for the set partitioning problem is a major goal of our research team.

## References

Ahuja RK, Magnanti TL, Orlin JB (1993) Network Flows: Theory, Algorithms, and Applications. Prentice Hall

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. Operations Research 46(3):316–329, doi:10.1287/opre.46.3.316

Ben Amor HMT, Desrosiers J, Frangioni A (2009) On the choice of explicit stabilizing terms in column generation. Discrete Applied Mathematics 157(6):1167–1184, doi:10.1016/j.dam.2008.06.021

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

Bland RG (1977) New finite pivoting rules for the simplex method. Mathematics of Operations Research 2(2):103–107, doi:10.1287/moor.2.2.103

Charnes A (1952) Optimality and degeneracy in linear programming. Econometrica 20(2):160–170, doi:10.2307/1907845

Dantzig GB (1963) Linear Programming and Extensions. Princeton University Press, New Jersey

Dantzig GB, Wolfe P (1960) Decomposition principle for linear programs. Operations Research 8(1):101–111, doi:10.1287/opre.8.1.101

Desaulniers G, Desrosiers J, Ioachim I, Solomon MM, Soumis F, Villeneuve D (1998) A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In: Crainic T, Laporte G (eds) Fleet Management and Logistics, Kluwer, pp 57–93, doi:10.1007/978-1-4615-5755-5_3

Desrosiers J, Dumas Y, Solomon MM, Soumis F (1995) Time constrained routing and scheduling. In: Ball M, Magnanti T, Monma C, Nemhauser G (eds) Handbooks in Operations Research and Management Science, Vol. 8: Network Routing, Elsevier B.V, pp 35–139, doi:10.1016/S0927-0507(05)80106-9

Desrosiers J, Gauthier JB, Lübbecke ME (2013a) A contraction-expansion algorithm for the capacitated minimum cost flow problem. Presentation at VeRoLog 2013, EURO Working Group on Vehicle Routing and Logistics Optimization, Southampton, UK

Desrosiers J, Gauthier JB, Lübbecke ME (2013b) Row-reduced column generation for degenerate master problems. Les Cahiers du GERAD G–2013–09, HEC Montréal, Canada du Merle O, Villeneuve D, Desrosiers J, Hansen P (1999) Stabilized column generation. Discrete Mathematics 194:229–237, doi:10.1016/S0012-365X(98)00213-1

Elhallaoui I, Villeneuve D, Soumis F, Desaulniers G (2005) Dynamic aggregation of set partitioning constraints in column generation. Operations Research 53(4):632–645, doi:10.1287/opre.1050.0222

Elhallaoui I, Desaulniers G, Metrane A, Soumis F (2008) Bi-dynamic constraint aggregation and subproblem reduction. Computers & Operations Research 35(5):1713–1724, doi:10.1016/j.cor.2006.10.007

Elhallaoui I, Metrane A, Soumis F, Desaulniers G (2010) Multi-phase dynamic constraint aggregation for set partitioning type problems. Mathematical Programming 123(2):345–370, doi:10.1007/s10107-008-0254-5

Elhallaoui I, Metrane A, Desaulniers G, Soumis F (2011) An Improved Primal Simplex algorithm for degenerate linear programs. INFORMS Journal on Computing 23:569–577, doi:10.1287/ijoc.1100.0425

Fukuda K (1982) Oriented matroid programming. PhD thesis, University of Waterloo, Ontario, Canada

Goldberg AV, Tarjan RE (1989) Finding minimum-cost circulations by canceling negative cycles. Journal of the ACM 36(4):873–886, doi:10.1145/76359.76368

Harris PMJ (1973) Pivot selection methods of the Devex LP code. Mathematical Programming 5(1):1–28, doi:10.1007/BF01580108

Karp RM (1978) A characterization of the minimum cycle mean in a digraph. Discrete Mathematics 23(3):309–311, doi:10.1016/0012-365X(78)90011-0

Klee V, Minty GJ (1972) How good is the simplex algorithm? In: Shisha O (ed) Inequalities, vol III, Academic Press, pp 159–175

Löbel A (1998) Vehicle scheduling in public transit and Lagrangean pricing. Management Science 44(12):1637–1649, doi:10.1287/mnsc.44.12.1637

Lübbecke ME, Desrosiers J (2005) Selected topics in column generation. Operations Research 53(6):1007–1023, doi:10.1287/opre.1050.0234

Mamer JW, McBride RD (2000) A decomposition-based pricing procedure for large-scale linear programs – An application to the linear multicommodity flow problem. Management Science 46(5):693–709, doi:10.1287/mnsc.46.5.693.12042

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

Oukil A, Ben Amor HMT, Desrosiers J, El Gueddari H (2007) Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems. Computers & Operations Research 34(3):817–834, doi:10.1016/j.cor.2005.05.011

Pan PQ (1998) A basis deficiency-allowing variation of the simplex method for linear programming. Computers & Mathematics with Applications 36(3):33–53, doi:10.1016/S0898-1221(98)00127-8

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

Raymond V, Soumis F, Metrane A (2009) Improved primal simplex version 3: Cold start, generalization for bounded variable problems and a new implementation. Les Cahiers du GERAD G–2009–15, HEC Montréal, Canada

Raymond V, Soumis F, Metrane A, Desrosiers J (2010a) Positive edge: A pricing criterion for the identification of non-degenerate simplex pivots. Les Cahiers du GERAD G–2010–61, HEC Montréal, Canada

Raymond V, Soumis F, Orban D (2010b) A new version of the Improved Primal Simplex for degenerate linear programs. Computers & Operations Research 37(1):91–98, doi:10.1016/j.cor.2009.03.020

Ryan DM, Osborne MR (1988) On the solution of highly degenerate linear programmes. Mathematical Programming 41(1-3):385–392, doi:10.1007/BF01580776

Sadykov R, Vanderbeck F (2013) Column generation for extended formulations. EURO Journal on Computational Optimization 1(1-2):81–115, doi:10.1007/s13675-013-0009-9

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

Towhidi M, Desrosiers J, Soumis F (2012) The positive edge pivot rule within COIN-ORs CLP. Les Cahiers du GERAD G–2012–77, HEC Montréal, Canada

Valério de Carvalho JM (1999) Exact solution of bin-packing problems using column generation and branch-and-bound. Annals of Operations Research 86(0):629–659, doi:10.1023/A:1018952112615

Valério de Carvalho JM (2002) LP models for bin-packing and cutting stock problems. European Journal of Operational Research 141(2):253–273, doi:10.1016/S0377-2217(02)00124-8

Wolfe P (1963) A technique for resolving degeneracy in linear programming. Journal of the Society for Industrial and Applied Mathematics 11(2):205–211, doi:10.1137/0111016