European Journal of Operational Research 236 (2014) 453–460

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Row-reduced column generation for degenerate master problems

Jacques Desrosiers [a,*], Jean Bertrand Gauthier [a], Marco E. Lübbecke [b]

[a] GERAD & HEC Montréal, 3000, Chemin de la Côte-Sainte-Catherine, Montréal, Québec H3T 2A7, Canada
[b] RWTH Aachen University, Chair of Operations Research, Kackertstraße 7, D-52072 Aachen, Germany

ARTICLE INFO

Article history:
Received 9 February 2013
Accepted 7 December 2013
Available online 17 December 2013

Keywords:
Column generation
Degeneracy
Dynamic row-reduction

ABSTRACT

Column generation for solving linear programs with a huge number of variables alternates between solving a master problem and a pricing subproblem to add variables to the master problem as needed. The method is known to often suffer from degeneracy in the master problem. Inspired by recent advances in coping with degeneracy in the primal simplex method, we propose a *row-reduced column generation* method that may take advantage of degenerate solutions. The idea is to reduce the number of constraints to the number of strictly positive basic variables in the current master problem solution. The advantage of this row-reduction is a smaller working basis, and thus a faster re-optimization of the master problem. This comes at the expense of a more involved pricing subproblem, itself eventually solved by column generation, that needs to generate weighted subsets of variables that are said compatible with the row-reduction, if possible. Such a subset of variables gives rise to a strict improvement in the objective function value if the weighted combination of the reduced costs is negative. We thus state, as a by-product, a necessary and sufficient optimality condition for linear programming.

This methodological paper generalizes the *improved primal simplex* and *dynamic constraints aggregation* methods. On highly degenerate linear programs, recent computational experiments with these two algorithms show that the row-reduction of a problem might have a large impact on the solution time. We conclude with a few algorithmic and implementation issues.

1. Introduction

Column generation, invented to solve large-scale linear programs (LPs), is particularly successful in the context of branch-and-price (Barnhart, Johnson, Nemhauser, Savelsbergh, & Vance, 1998; Lübbecke & Desrosiers, 2005) for solving well-structured integer programs. Column generation is used to solve the LP relaxations at each node of a search tree, and often produces strong dual bounds. It alternates between solving a restricted master problem (an LP) and one or several subproblems (usually integer programs) in order to dynamically add new variables to the model. Like the primal simplex method, column generation suffers from degeneracy. This defect is particularly visible when solving LP relaxations of combinatorial optimization problems, a main application area of branch-and-price.

In this methodological paper, we present a *row-reduced column generation* (RrCG) method which turns degeneracy into a potential advantage. Our method differs from standard column generation only in iterations where degeneracy occurs in the restricted master problem (RMP). We dynamically partition the RMP constraints based on the numerical values of the current basic variables. The idea is to keep only those constraints in the RMP that correspond to strictly positive basic variables. This leads to a *row-reduced* restricted master problem which does not only discard most variables from consideration in column generation, but also reduces the number of constraints, and in particular the size of the current working basis. In linear algebra terms, we work with a projection into the subspace spanned by the column-vectors of the non-degenerate variables. This is similar to the idea of a deficient basis in the simplex method (Pan, 1998). This row reduction comes at the expense of a computationally more involved pricing subproblem which needs itself to be solved by column generation.

Degeneracy in column generation has been dealt with using perturbation of the right-hand side as in the standard primal simplex method. In particular, certain dual variable stabilization approaches explicitly use perturbation, see Ben Amor, Desrosiers, and Frangioni (2009) for a stabilized column generation framework and the many references therein. Recently, a new line of research emerged for coping with primal degeneracy in linear programming, namely the ***improved primal simplex*** method (IPS) (Elhallaoui, Metrane, Desaulniers, & Soumis, 2010; Raymond, Soumis, & Orban, 2010). Our work generalizes IPS for solving degenerate linear programs and the *dynamic constraints aggregation* method (Elhallaoui, Desaulniers, Metrane, & Soumis, 2008; Elhallaoui,

* Corresponding author. Tel.: +1 514 340 6505; fax: +1 514 340 5634.
E-mail addresses: jacques.desrosiers@hec.ca (J. Desrosiers), jean-bertrand.gauthier@hec.ca (J.B. Gauthier), marco.luebbecke@rwth-aachen.de (M.E. Lübbecke).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.12.016

Villeneuve, Soumis, & Desaulniers, 2005; Elhallaoui, Metrane, Soumis, & Desaulniers, 2010) for solving LP relaxations of set partitioning problems (by column generation) stemming from vehicle routing and crew scheduling applications. The referenced papers suggest that, on highly degenerate linear programs, a row-reduction of a problem shows great promise in reducing overall solution times.

The paper is organized as follows. Section 2 recalls the column generation method with the definitions of the master problem MP, its variable-restricted version RMP and its pricing subproblem SP. Section 3 presents the RrCG approach. It essentially defines the row and column partitions of the master problem based on a current degenerate solution, introduces the *row-reduced restricted master problem* RrRMP and its associated pricing subproblem rSP, and finally brings in a specialized column generator cSP for single columns compatible with the row-reduced master problem. Section 3.4 discusses the case of inequality constraints followed by an algorithm. Section 3.6 provides a necessary and sufficient optimality condition for linear programs. Finally, Section 4 discusses some properties followed by implementation issues. Our conclusions complete the paper in Section 5.

2. Column generation

Let us briefly recall the mechanism of standard column generation, see Lübbecke and Desrosiers (2005) for a general introduction. We would like to solve the following linear program, called the *master problem* (MP), with a prohibitively large number of variables $\boldsymbol{\lambda} \in \mathbb{R}^n_+$

$$z^\star_{MP} := \begin{array}{lllll} \min & \mathbf{c}^\intercal\boldsymbol{\lambda} & & & \\ \text{s.t.} & A\boldsymbol{\lambda} & = & \mathbf{b} & [\boldsymbol{\pi}] \\ & \boldsymbol{\lambda} & \geqslant & \mathbf{0}, & \end{array} \tag{1}$$

where $A \in \mathbb{R}^m \times \mathbb{R}^n, \mathbf{c} \in \mathbb{R}^n$, and $\mathbf{b} \in \mathbb{R}^m$. The corresponding dual variables $\boldsymbol{\pi} \in \mathbb{R}^m$ are listed in brackets. We assume that $\boldsymbol{\lambda}$ includes m non-negative artificial variables, hence A is of full row rank, and MP is feasible if $\mathbf{b} \geqslant \mathbf{0}$. In applications, every coefficient column $\mathbf{a}$ of A encodes a combinatorial object $\mathbf{x} \in X$ like a path, permutation, set, or multi-set. To stress this fact, we write $\mathbf{a} = \mathbf{a}(\mathbf{x})$ and $c = c(\mathbf{x})$ for its cost coefficient. Column generation works with a *restricted master problem* (RMP) which involves a small subset of variables only. At each iteration, RMP is solved to optimality first. Then, like in the primal simplex algorithm, we look for a non-basic variable to price out and enter the current basis. That is, we either find a column $\mathbf{a}(\mathbf{x})$ of cost $c(\mathbf{x})$ with a negative reduced cost $\bar{c}(\mathbf{x})$ or need to prove that no such variable exists. This is accomplished by solving the *pricing subproblem* (SP)

$$\bar{c}^\star_{SP} := \min_{\mathbf{x}\in\mathbf{X}}\{c(\mathbf{x}) - \boldsymbol{\pi}^\intercal\mathbf{a}(\mathbf{x})\}. \tag{2}$$

If $\bar{c}^\star_{SP} \geqslant 0$, no negative reduced cost columns exist and the current solution $\boldsymbol{\lambda}$ of RMP (embedded into $\mathbb{R}^n_+$) optimally solves MP (1) as well. Otherwise, a minimizer of SP (2) gives rise to a variable to be added to RMP, and we iterate.

Functions $c(\mathbf{x})$ and $\mathbf{a}(\mathbf{x})$ may be linear functions, as in a Dantzig-Wolfe reformulation of a linear program (Dantzig & Wolfe, 1960), but $c(\mathbf{x})$ is typically non-linear in many practical applications such as in rich vehicle routing and crew scheduling (Desaulniers et al., 1998). Functions $\mathbf{a}(\mathbf{x})$ are also non-linear when Chvátal-Gomory cuts are derived from the master problem variables, see Desaulniers, Desrosiers, and Spoorendonk (2011). Non-linearities may increase the difficulty in solving SP, but it always ends up in a scalar cost c_j and a vector $\mathbf{a}_j$ of scalar coefficients for each variable λ_j in MP, $j \in \{1,\dots,n\}$.

2.1. Notation

Vectors are written in bold face. We denote by I_k the $k \times k$ identity matrix and by $\mathbf{0}$ (resp. $\mathbf{1}$) a vector/matrix with all zero (resp. one) entries of appropriate contextual dimensions. For subsets $I \subseteq \{1,\dots,m\}$ of row-indices and subsets $J \subseteq \{1,\dots,n\}$ of column-indices we denote by A_{IJ} the sub-matrix of A containing the rows and columns indexed by I and J, respectively. We further use standard linear programming notation like $A_J\boldsymbol{\lambda}_J$, the subset of columns of A which are indexed by J multiplied by the corresponding sub-vector of variables $\boldsymbol{\lambda}_J$. There is one notable exception: The set N will *not* denote the non-basis (but usually a superset). Even though one never actually computes the inverse of a basis matrix, our exposition will sometimes rely on"tableau data," when it is conceptually more convenient.

3. Row-reduced column generation

RMP is a *column-reduced* MP and its variables are generated as needed by solving SP. The *row-reduced column generation* comes into play when the current solution of RMP is degenerate with $p < m$ positive variables. In what follows, we define a *row-reduced* RMP, denoted RrRMP, which decreases the number of rows to only p. The case with equality constraints is treated first as formulated in (1), and a generalization to the inequality-constrained case is presented in Section 3.4.

3.1. Row and column partitions

Let $\boldsymbol{\lambda}$ be a feasible solution to MP, with the index set $F \subset \{1,\dots,n\}$ of variables at strictly positive value, that is, $\boldsymbol{\lambda}_F > \mathbf{0}$. These variables are *free* to increase or decrease relatively to their current values. All other possibly present variables assume a *null* value, that is, $\boldsymbol{\lambda}_N = \mathbf{0}$ for $N := \{1,\dots,n\} \setminus F$. We assume that $\boldsymbol{\lambda}$ is degenerate in the sense that the number of positive variables is less than the number of rows of MP, i.e., $|F| = p < m$. The columns of A_F are required to be linearly independent, which is no restriction when $\boldsymbol{\lambda}$ is computed with a simplex algorithm. This assumption allows us to construct a basis matrix A_B for MP representing the solution $\boldsymbol{\lambda}$ in the following way. Identify a subset $P \subset \{1,\dots,m\}$ of p linearly independent rows of A_F and "fill up" with $m-p$ unit columns to provide for artificial basic variables in the rows indexed by $Z := \{1,\dots,m\} \setminus P$. More precisely, this yields the following form

$$A_B = \begin{bmatrix} A_{PF} & \mathbf{0} \\ A_{ZF} & I_{m-p} \end{bmatrix}. \tag{3}$$

One way to accomplish this form is to initialize the RMP with columns A_F and m artificial variables and apply a *phase I* of the primal simplex algorithm. The above construction induces row and column partitions, and MP (and the corresponding vector of dual variables $\boldsymbol{\pi}$) reads as

$$z^\star_{MP} := \begin{array}{lllllll} \min & \mathbf{c}_F^\intercal\boldsymbol{\lambda}_F & + & \mathbf{c}_N^\intercal\boldsymbol{\lambda}_N & & & \\ \text{s.t.} & A_{PF}\boldsymbol{\lambda}_F & + & A_{PN}\boldsymbol{\lambda}_N & = & \mathbf{b}_P & [\boldsymbol{\pi}_P] \\ & A_{ZF}\boldsymbol{\lambda}_F & + & A_{ZN}\boldsymbol{\lambda}_N & = & \mathbf{b}_Z & [\boldsymbol{\pi}_Z] \\ & \boldsymbol{\lambda}_F, & & \boldsymbol{\lambda}_N & \geqslant & \mathbf{0}. & \end{array} \tag{4}$$

The inverse of the above basis matrix (3) has a particularly easy form,

$$A_B^{-1} = \begin{bmatrix} A_{PF}^{-1} & \mathbf{0} \\ -A_{ZF}A_{PF}^{-1} & I_{m-p} \end{bmatrix}. \tag{5}$$

If we left-multiply (4) by A_B^{-1}, we obtain the equivalent "tableau data" formulation