



Discrete Optimization

All-integer column generation for set partitioning: Basic principles and extensions



Elina Rönnberg*, Torbjörn Larsson

Department of Mathematics, Linköping University, SE-581 83 Linköping, Sweden

ARTICLE INFO

Article history:

Received 21 March 2012

Accepted 25 August 2013

Available online 5 September 2013

Keywords:

Integer programming

Column generation

Set partitioning

Quasi-integrality

All-integer pivots

ABSTRACT

Column generation, combined with an appropriate integer programming technique, has shown to be a powerful tool for solving huge integer programmes arising in various applications. In these column generation approaches, the master problem is often of a set partitioning type.

The set partitioning polytope has the quasi-integrality property, which enables the use of simplex pivots for finding improved integer solutions, each of which is associated with a linear programming basis. By combining such pivots with column generation, one obtains a method where each found solution to a restricted master problem is feasible, integer, and associated with a dual solution that can be used in a column generation step.

This paper presents a framework for such an all-integer column generation approach to set partitioning problems. We give the basic principles of all-integer pivots and all-integer column generation. We also state optimality conditions and introduce means for preserving a basis in the event that a heuristic is applied to the master problem. These extensions introduce flexibility in the design of a specific solution scheme of this kind, and with proper settings optimal or approximate solutions can be sought for.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Column generation is a linear programming method that is designed for solving problems that have huge numbers of variables, but also a certain structure. When complemented with a suitable integer programming technique, column generation has proven to be successful in solving many large scale integer programmes. A well known strategy of this type is branch-and-price, where column generation is embedded in a branch-and-bound scheme, see Barnhart, Johnson, Nemhauser, Savelsbergh, and Vance (1998) and Vanderbeck (2011). For surveys of column generation and applications within integer programming, see Lübbecke and Desrosiers (2005) and Wilhelm (2001). Our work originates from the fact that many applications of column generation give rise to master problems of a set partitioning type, as for example aircrew rostering problems, see Gamache, Soumis, Marquis, and Desrosiers (1999), and the generalised assignment problem, see Barnhart et al. (1998).

The feasible set of the linear programming relaxation of the set partitioning problem possesses the *quasi-integrality property*, which means that for the polytope of feasible solutions, every edge of the convex hull of the integer points is also an edge of the polytope itself. This property makes it possible to utilise linear

programming techniques for finding improved integer solutions, an idea which was first explored by Trubin (1969). It was later somewhat more thoroughly described and named *the integral simplex method* in Yemelichev, Kovalev, and Kravtsov (1984). The integral simplex method only acts as a local search method, but complemented with a branching strategy, as in the implementation made by Thompson (2002), the finding of an optimal solution can be ensured. In Rönnberg and Larsson (2009), based on results in Balas and Padberg (1972), it is shown how the column generation principle can be integrated with the integral simplex method, and how a branching strategy can be used to ensure the finding of an optimal solution.

The work to be presented in this paper is a continuation of the research published in Rönnberg and Larsson (2009), Rönnberg (2012, 2008), although we here consider more general pivots than those used in the integral simplex method and Rönnberg and Larsson (2009); these are called *all-integer pivots*. We develop the basic principles of *all-integer column generation* for set partitioning problems, together with several extensions.

To be more precise, we consider column generation problems that are consistent with all-integer pivots, state optimality conditions for the integer problem and use the concept of *over-generation of columns*, which means generating not only an optimal column but several, by generating near-optimal solutions to a regular column generation subproblem. This can be used as an additional way to augment the master problem, or for generating

* Corresponding author. Tel.: +46 13 281645.

E-mail address: elron@mai.liu.se (E. Rönnberg).

enough columns for guaranteeing that the master problem can provide an integer optimal solution.

Further, so called *surrogate columns* are introduced. These facilitate the use of combinatorial heuristics on the master problem within a framework of all-integer column generation. This feature is of interest because strategies for how to combine column generation and heuristics are not very well explored in the literature, although the development of such strategies seem to attract more attention; see for example (Joncour, Michel, Sadykov, Sverdllov, & Vanderbeck, 2010), which gives an overview, and also the Search-Col project (Metaheuristic Search by Column Generation, 2013) on this topic.

Although the overall approach is primarily introduced as being of a metaheuristic nature, criteria for determining whether a solution is optimal or near-optimal are also available. Key characteristics of the approach are that all successively found solutions are feasible and integral, and that the integrality requirement on the variables are handled differently from what has previously been described in the column generation literature.

The outline of the paper is as follows. Section 2 introduces notations and other prerequisites. Section 3 presents the foundations and principles of all-integer column generation for set partitioning: the quasi-integrality property, all-integer pivots, and all-integer column generation. The pivoting step naturally separates into the non-degenerate and degenerate cases, which aim at a primal and a dual improvement of the solution respectively. The same cases hold for the column generation step, and we show how the all-integer column generation subproblem can be tailored to produce columns enabling either non-degenerate or degenerate pivots.

Section 4.1 derives the surrogate column pivot, which enables making an all-integer pivot from an integer feasible solution with a known basis, to another integer feasible solution for which no basis is known. Section 4.2 gives optimality conditions for the integer problem and outlines the over-generation strategy. In Section 4.3, the generalised assignment problem is used to illustrate the formulation of the all-integer column generation subproblems and how to transform the subproblems into a more suitable form. This transformation is believed to be of critical importance from a computational point of view. Also presented in this section are some simple computational experiments that illustrate the behaviour of the core components of the all-integer column generation approach. Some reflections on the overall methodology follows in Section 5, together with some ideas for future work.

Before proceeding with our presentation we would like to mention that one of the reviewers drew our attention to an interesting parallel work of Zaghrouti et al. (2013). The similarity to our work is that they utilise the quasi-integrality property to come up with an efficient way to generate linear programming pivots leading to integer extreme points. For this purpose they use a direction-finding subproblem that can be solved efficiently and they present excellent computational results for set partitioning problems with up to 500,000 variables. The main difference is that (Zaghrouti et al., 2013) do not consider a column generation setting.

2. Problem formulation and notation

The set partitioning problem to be solved is

$$\begin{aligned}
 \text{[SPP}_{\mathcal{N}}\text{]} \quad z^* &= \min \sum_{j \in \mathcal{N}} c_j x_j \\
 \text{s.t.} \quad &\sum_{j \in \mathcal{N}} a_{ij} x_j = e_i, \quad i \in M \\
 &x_j \in \{0, 1\}, \quad j \in \mathcal{N}.
 \end{aligned}$$

Here, \mathcal{N} and $M = \{1, \dots, m\}$ are the sets of indices for the variables respective constraints. Further, $c_j \in \mathbf{R}$, $j \in \mathcal{N}$, $a_{ij} \in \{0, 1\}$, $i \in M$, $j \in \mathcal{N}$, and $e_i = 1$, $i \in M$. We assume that $|\mathcal{N}| \geq m$ and that the problem is feasible. Let $a_j = (a_{1j}, \dots, a_{mj})^T$, $j \in \mathcal{N}$, and assume, with no loss of generality, that the matrix $(a_1, \dots, a_{|\mathcal{N}|})$ has no zero column and full rank.

In a column generation setting, problem $\text{SPP}_{\mathcal{N}}$ is referred to as the complete master problem. Let $\mathcal{P} = \{(c_j, a_j) : j \in \mathcal{N}\} \subseteq Z \times \{0, 1\}^m$ be the set of columns. This set typically corresponds to the set of feasible solutions to an optimisation problem and it is typically huge. In an application there might be columns that are identical except for their costs, and in such cases, only the cheapest one will be contained in \mathcal{P} ; hence $a_j \neq a_k$ whenever $j \neq k$.

Let SPP_N be a restricted set partitioning problem, or restricted master problem, where the variables with indices $N = \{1, \dots, n\} \subset \mathcal{N}$ are at hand. It is assumed that $n \geq m$, that SPP_N is feasible, and that the matrix $A = (a_1, \dots, a_n)$ has full rank. An optimal solution and the optimal objective value of SPP_N are denoted by \bar{x} and \bar{z} , respectively.

Let SPP_N^{LP} be the linear programming relaxation of SPP_N , with $x_j \in \{0, 1\}$ replaced by $x_j \geq 0$, $j \in N$. Because the feasible set is contained within the unit hypercube, all feasible integer points will be extreme points of the polytope describing the feasible set of SPP_N^{LP} . For an extreme point to SPP_N^{LP} , denote a basis by $B \subseteq \{a_1, \dots, a_n\}$; if an extreme point is degenerate, it is associated with more than one basis. For a given basis, let I and J denote the corresponding sets of basic and non-basic columns respectively, let $x_B = (x_j)_{j \in I}$ and $x_N = (x_j)_{j \in J}$ be the corresponding basic and non-basic components of x , and let $Q \subseteq I$ be the set of basic variables that take the value one. The notations B, I, J and Q will sometimes be used together with a superscript referring to the extreme point with which they are associated. Further, let $\bar{e} = B^{-1}e$, where $e = (e_1, \dots, e_m)^T$, and $\bar{a}_j = B^{-1}a_j$, $j \in N$, be the updated right-hand-side and constraint columns respectively. The complementary dual solution is $u^T = c_B^T B^{-1}$, where $c_B = (c_j)_{j \in I}$, and the reduced costs are $\bar{c}_j = c_j - u^T a_j$, $j \in J$.

For SPP_N^{LP} , with basis B and dual solution u , the column generation subproblem

$$\begin{aligned}
 \text{[CG]} \quad \bar{c}_p &= \min \quad c - \sum_{i \in M} u_i a_i \\
 \text{s.t.} \quad &(c, a) \in \mathcal{P},
 \end{aligned}$$

is used to produce a column (c_p, a_p) that can be added to the restricted master problem. For future reference, we note that if the column generation is made using a dual feasible solution with respect to the problem SPP_N^{LP} , then the generated column (c_p, a_p) is never already included in the restricted master problem SPP_N (unless dual feasibility holds also in $\text{SPP}_{\mathcal{N}}$, in which case $\bar{c}_p = 0$). If however, the column generation is made using a dual infeasible solution with respect to SPP_N^{LP} , then it might happen that the column (c_p, a_p) is already included in the restricted master problem.

3. Background

This section presents the theoretical background needed for the results to be presented in this paper. The first results are on the polyhedral structure of the set partitioning problem and trace all the way back to the sixties (Trubin, 1969), followed by recent results from Rönnberg and Larsson (2009) that introduce a way of utilising this polyhedral structure in a column generation setting.

3.1. Quasi-integrality and all-integer pivots

One way of finding an improved solution to the restricted master problem is to use the integrality preserving pivots to be

Download English Version:

<https://daneshyari.com/en/article/6897665>

Download Persian Version:

<https://daneshyari.com/article/6897665>

[Daneshyari.com](https://daneshyari.com)