Computers & Operations Research 40 (2013) 2026–2036

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Using the primal-dual interior point algorithm within the branch-price-and-cut method

Pedro Munari [a,*,1], Jacek Gondzio [b]

[a] Instituto de Ciências Matemáticas e de Computação, University of São Paulo, Av. Trabalhador, São-carlense, 400 - Centro, Cx. Postal 668, CEP 13560-970, São Carlos-SP, Brazil

[b] School of Mathematics, The University of Edinburgh, James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, United Kingdom

ARTICLE INFO

Available online 14 March 2013

Keywords:
Branch-price-and-cut
Column generation
Primal-dual interior point algorithm
Vehicle routing problem

ABSTRACT

Branch-price-and-cut has proven to be a powerful method for solving integer programming problems. It combines decomposition techniques with the generation of both columns and valid inequalities and relies on strong bounds to guide the search in the branch-and-bound tree. In this paper, we present how to improve the performance of a branch-price-and-cut method by using the primal-dual interior point algorithm. We discuss in detail how to deal with the challenges of using the interior point algorithm with the core components of the branch-price-and-cut method. The effort to overcome the difficulties pays off in a number of advantageous features offered by the new approach. We present the computational results of solving well-known instances of the vehicle routing problem with time windows, a challenging integer programming problem. The results indicate that the proposed approach delivers the best overall performance when compared with a similar branch-price-and-cut method which is based on the simplex algorithm.

1. Introduction

The branch-price-and-cut method has been widely used for solving integer programming models in which a special structure can be identified in the coefficient matrix. This structure is exploited by a reformulation technique, *e.g.* the Dantzig–Wolfe decomposition, that usually leads to a formulation with a stronger linear relaxation when compared with the linear relaxation of the original model. By using this stronger formulation within a branch-and-bound tree, we obtain the branch-and-price method. Since the reformulation may have a huge number of variables, a column generation algorithm is used to solve the linear relaxation. For this reason, the branch-and-price method is also known as integer programming column generation. In some cases, valid inequalities should also be added to the reformulated model to get even better bounds from the linear relaxations with the aim of improving the branch-and-bound search, which leads to the branch-price-and-cut method. See [1–3] for comprehensive surveys on these methods.

In the vast majority of branch-price-and-cut implementations presented in the literature, a simplex-type method is used to solve the linear programming problems at each node. Hence, the generation of columns and valid inequalities are typically based on optimal solutions of these problems. Particularly in the column generation case, the use of optimal dual solutions which are extreme points of the dual feasible set may adversely affect the performance of the method. The reason is the high oscillation between extreme points from one iteration to another, which may result in slow convergence as well as a temporary stalling of the algorithm, specially in highly degenerate solutions. To overcome this behavior, different strategies have been proposed in the literature which use non-extremal dual solutions so that more stable column generation procedures are obtained [4–9]. Other strategies, such as dynamic constraint aggregation, have also been used with the same purpose [10]. Using non-extremal primal solutions to generate valid inequalities has shown to be more effective as well, since deeper cuts are obtained and a smaller number of them are usually needed [11–14]. In this paper, we investigate the use of the primal-dual interior point method (PDIPM) to provide primal and dual non-optimal solutions which are well-centered in the feasible set. The computational experience provides evidence that these solutions are beneficial to the generation of columns and valid inequalities.

Although very successful in many other fields related to linear programming, interior point methods do not seem to have made a big impact in the integer programming context. It is probably because the standard integer programming methodologies were originally proposed at time when the simplex method was the only efficient algorithm available for solving linear programming

* Corresponding author. Tel.: +55 16 33519525.
E-mail addresses: munari@icmc.usp.br (P. Munari), J.Gondzio@ed.ac.uk (J. Gondzio).

[1] Supported by FAPESP (São Paulo Research Foundation, Brazil) under the projects 2008/09040-7 and 2012/05486-6.

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.02.028

problems and, hence, they were biased to the features available in this method. Moreover, until a few years ago, interior point methods were not able to reoptimize a problem after carrying out modifications to the data as efficiently as a simplex type method, a scenario that has changed in the last years with the development of efficient warmstarting techniques for interior point methods [15–17]. Another reason may be due to previous unsuccessful attempts of straightforwardly replacing a simplex type method by an interior point method. These two methods are very different [18] and, hence, should not be used in the same way.

In this paper, we address the several facets of using the PDIPM within a branch-price-and-cut method. We discuss in detail how to modify the core components of this method in order to exploit the advantageous features offered by the interior point algorithm. We believe it is an appropriate time for an investigation like this, as interior point methods have achieved a mature status concerning both theory and computational implementations. To verify the proposed approach, we present computational results for well-known instances of the vehicle routing problem with time windows (VRPTW). It is a classical application of integer programming that is widely used for testing new algorithms due to its difficulty. It is worth mentioning that the issues related to the integration of the interior point algorithm with the branch-price-and-cut method which we discuss in this paper are not limited to or specialized for the VRPTW and can be straightforwardly used in other combinatorial optimization applications.

The remainder of this paper is organized as follows. In Section 2, we briefly describe the fundamental concepts of the PDIPM and present a literature review of previous attempts of combining interior point algorithms with integer programming methodologies. In Section 3, we address all the issues involved in the use of the PDIPM in the branch-price-and-cut method and propose how to deal with them. The VRPTW is briefly described in Section 4 and the results of computational experiments with the new approach are presented in Section 5. The conclusion and potential further developments are presented in Section 6.

2. Interior point methods and integer programming

Starting with Karmarkar's projective algorithm [19], interior point methods have quickly and strongly evolved in the last few decades [18,20]. They have been successfully applied not only to solving linear programming problems but also in many other areas, such as quadratic, semi-definite and conic programming. However, in spite of the close relationship between linear programming and integer programming, interior point methods have not showed a similar impact on integer programming. In this section we briefly describe the fundamental concepts of the PDIPM and then present a literature review of different attempts to use interior point methods within integer programming approaches that rely on linear programming relaxations.

2.1. The primal-dual interior point method (PDIPM)

In this paper, we consider a linear programming problem represented by the following primal (P) and dual (D) formulations:

$$\begin{aligned}
(P)\quad & \min\ c^T\lambda \\
& \text{s.t.}\ A\lambda = b, \\
& \lambda \ge 0, \\
(D)\quad & \max\ b^T u \\
& \text{s.t.}\ A^T u + s = c, \\
& s \ge 0,
\end{aligned} \tag{2.1}$$

where A is a full row rank matrix of columns $a_j \in \mathbb{R}^m$, $j \in N = \{1,\dots,n\}$, $\lambda \in \mathbb{R}^n$ is the vector of primal variables, $u \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ are the vectors of dual variables, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ are parameters of the problem, and $0 < m \le n$. The first order optimality conditions associated to this pair of problems are given by

$$b - A\lambda = 0, \tag{2.2}$$

$$c - A^T u - s = 0, \tag{2.3}$$

$$\Lambda S e = 0, \tag{2.4}$$

$$(\lambda, s) \ge 0, \tag{2.5}$$

where $\Lambda = \mathrm{diag}(\lambda_1,\dots,\lambda_n)$, $S = \mathrm{diag}(s_1,\dots,s_n)$ and $e = (1,1,\dots,1)$. The PDIPM is based on the perturbation of the optimality conditions, in which Eq. (2.4) is replaced by $\Lambda Se = \mu e$ where $\mu > 0$ is the *barrier parameter*, or *duality measure*. This parameter is gradually driven to zero throughout the iterations so that a primal-dual pair of optimal solutions satisfying (2.2)–(2.5) is obtained at the end of the algorithm. For a given value of μ, the perturbed optimality conditions have a unique solution, which is called a μ-center. The set composed of all μ-centers is called a central-path. Instead of strictly satisfying the perturbed optimality conditions, the iterates of the PDIPM belong to a neighborhood of the central path. The idea of the neighborhood is to keep the iterates well-centered and in a safe area so that all variables approach their optimal values with a uniform pace. Different neighborhoods have been proposed in the literature. For instance, the symmetric neighborhood requires all the iterates to satisfy (2.2) and (2.3) and further belong to the set

$$\{(\lambda,u,s) \mid \gamma\mu \le \lambda_j s_j \le (1/\gamma)\mu,\ \forall j = 1,\dots,n\},$$

for a given parameter $\gamma \in (0,1)$.

In a given iteration of the PDIPM, the corresponding iterate (λ,u,s) is modified by using the directions $(\Delta\lambda,\Delta u,\Delta s)$ which are obtained by solving the following Newton step equations:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & \Lambda \end{bmatrix}\begin{bmatrix} \Delta\lambda \\ \Delta u \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_\lambda \\ \xi_s \\ \sigma\mu e - \Lambda S e \end{bmatrix}, \tag{2.6}$$

where σ is a parameter used to reduce the complementarity gap $\lambda^T s$ of the next iterate, and vectors $\xi_\lambda = b - A\lambda$ and $\xi_s = c - A^T u - s$ are the primal and dual infeasibilities of the current iterate, respectively. After obtaining the directions, the step-sizes α_P and α_D are computed for the primal and dual components, respectively, in order to guarantee that the next iterate belongs to the neighborhood of the central path. The next iterate is then given by $(\lambda + \alpha_P\Delta\lambda, u + \alpha_D\Delta u, s + \alpha_D\Delta s)$. For a full description of the method and a discussion about its main theoretical and implementation issues, the reader is referred to [18].

2.2. Interior point methods within integer programming methodologies

Few attempts have been presented in the literature regarding integer programming methodologies that are based on interior point algorithms. The first implementations in this sense started only in the beginning of the 1990s, with the pioneering works by Mitchell and his collaborators. In [11], the primal projective interior point algorithm is combined with a cutting plane method and employed to solve the perfect matching problem. The authors propose to use early termination when solving the linear programming problems and present how to deal with several issues, such as how to obtain a new iterate after adding constraints and columns to a problem that has just been solved. As shown in the