Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 3, N. 1, 2015.
Trabalho apresentado no XXXV CNMAC, Natal-RN, 2014.

Column Generation and Branch-and-Price with Interior Point Methods*

Pedro Munari
Production Engineering Department, Federal University of São Carlos, Brazil
E-mail: munari@dep.ufscar.br

Jacek Gondzio
School of Mathematics, The University of Edinburgh, United Kingdom
E-mail: J.Gondzio@ed.ac.uk

Abstract: *Column generation and branch-and-price methods are currently essential tools for solving many classes of integer programming and combinatorial optimization problems. This paper addresses how to improve the performance of these methodologies by using the interior point algorithm. The purpose is to summarize the main developments proposed in the thesis "Theoretical and computational issues for improving the performance of linear optimization methods" which has been awarded with the "Doctoral Prize Odelar Leite Linhares" in 2014. As the computational experiments indicates, the interior point method is more than an alternative to the simplex method. Indeed, it offers advantageous features which can be used to stabilize the column generation technique and improve the overall performance of the branch-and-price method.*

Keywords: *column generation, branch-and-price, interior point algorithm, stabilization.*

1 Introduction

The integer programming methodologies have evolved significantly in the last decades. Currently, computer codes based on these methodologies are able to solve large-scale challenging problems which no one could imagine to solve a few decades ago. The column generation (CG) technique is an example of a successful method which was first used in the end of the Fifties [8] and then started a research area which has been very active [15]. The researches in this context have focused not only on applying CG to solve optimization problems, but also on ways to improve this methodology in order to solve problems faster as well as to solve more challenging problems. The combination of CG with the branch-and-bound method results in the branch-and-price (BP) method [4], which is paramount for solving many classes of integer programming problems, such as vehicle routing, cutting stock and lot sizing problems.

The CG technique was born from the need of improving the performance of the simplex method. Indeed, as reported by Ford and Fulkerson [8], the purpose of CG was to avoid the full enumeration of columns when solving a formulation of the multi-commodity network flow problem in which "*the number of variables was too large to be dealt with explicitly*". Since in a given iteration of the simplex method it is enough to consider only the variables which are currently basic, the remaining variables in the problem were treated implicitly by modifying the usual pricing operation. Instead of explicitly computing the reduced costs of each non-basic variables, the authors proposed to solve a subproblem which was able to *generate* a non-basic column with the smallest reduced cost. This modification led to the CG technique.

Possibly due to its origin in the simplex method, the CG technique is still strongly linked with this method. However, there is no need to link these two approaches, specially for the

*This research was supported by FAPESP through grant numbers 2008/09040-7 and 2012/05486-6.

following two main reasons. First, the CG technique is the dual counterpart of the cutting plane method [14], which has originally no link with the simplex method. Second, it has been shown in the literature that much can be gained in efficiency by using non-optimal dual solutions within CG [12, 10, 17].

Similarly, most implementations of *branch-and-price* and *branch-price-and-cut* methods rely on the simplex method [15, 4]. Hence, columns and valid inequalities are generated by using optimal solutions which correspond to extreme points. Branching is also performed by using optimal solutions in general. Again, there is no need to use the simplex method in this context. Actually, optimal solutions should rather be avoided when generating columns and valid inequalities, if well-centered suboptimal solutions are available [21].

In this paper we reinforce that the interior point method not only can, but should be used within CG and BP methodologies. Even though earlier efforts in this direction have been presented in the literature (see e.g. [17, 13, 16, 5, 18, 7, 21]), it is still usual to find papers in which the authors have tried using the interior point algorithm in this context but have reported very limited success. Such unsuccessful attempts are probably justified by the fact that using the interior point method within CG and BP is not straightforward. Indeed, this task should not be understood as simply replacing the simplex method. These two linear programming methodologies differ significantly and offer different advantages. When using each of them we should exploit the best they have to offer. Therefore, we aim at describing the main challenges regarding these combinations and the ways of using the advantageous features offered by the interior point method. By presenting computational experiments we provide evidence that using an interior point method is a natural way of stabilizing the CG technique and improving the overall performance of the BP method. The experiments were run by using the PDCGM library which is publicly available for download at: `http://www.maths.ed.ac.uk/~gondzio/software/pdcgm.html`.

The remainder of this paper is organized as follows. Section 2 describes the primal-dual column generation method, which is an efficient way of combining CG and the primal-dual interior point algorithm. Section 3 shows how the primal-dual interior point algorithm can be used to improve the performance of the BP method. In these two sections, we present the results of experiments with different classes of problems to illustrate the benefits of employing the primal-dual interior point method. Finally, Section 4 presents the concluding remarks.

2 Column Generation with the Interior Point Method

We are interested in solving a linear programming problem with a huge number of variables (columns), which we call *master problem* (MP). Assume that we know a rule which can be used to generate all the columns in the MP. Even if it was possible to enumerate all these columns, it would be very inefficient to solve the MP if we explicitly consider them. Instead, we may solve the problem iteratively by using the *CG technique*. First, we solve a reduced version of the MP in which only a relatively small subset of columns is explicitly available. We call this reduced version a *restricted* master problem (RMP), which can be represented as follows

$$z_{RMP} := \min \quad \sum_{j \in \overline{N}} c_j \lambda_j, \tag{1a}$$

$$\text{s.t.} \quad \sum_{j \in \overline{N}} a_j \lambda_j = b, \tag{1b}$$

$$\lambda_j \geq 0, \qquad \forall j \in \overline{N}, \tag{1c}$$

where $\lambda_j \in \mathbb{R}$ are called the master variables; parameters $c_j \in \mathbb{R}$ and $a_j \in \mathbb{R}^m$ are the cost and the coefficients of the j-th column in the problem, respectively; $b \in \mathbb{R}^m$ is the independent vector; and $\overline{N}$ is the set of indices of columns which have been explicitly generated. The optimal solution of the RMP is also an optimal solution of its corresponding MP, only if we cannot generate any

other column with a negative reduced cost. To verify that without explicitly generating new columns, we solve the following *pricing subproblem*:

$$z_{SP}(\overline{u}) := \min\{0; c_j - \overline{u}^T a_j \mid (c_j, a_j) \in \mathcal{C}\}, \tag{2}$$

where $\bar{u} \in \mathbb{R}^m$ is a dual solution associated to constraints (1b) and $\mathcal{C}$ is the set of all columns which can be generated for the MP. Notice that (c_j, a_j) is the vector of variables in the subproblem. By using the dual solution $\bar{u}$, the pricing subproblem must be able to generate at least one new column with a negative reduced cost, in case any exists. The new column is then added to the RMP which should be reoptimized, leading to a new iteration of the CG technique. In case no column with a negative reduced cost can be generated, the optimal solution of the current RMP is also an optimal solution of the MP and, therefore, the CG procedure terminates.

Combining the CG technique with the interior point algorithm is not straightforward. To be effective, this combination should involve more than just replacing the simplex method with the interior point algorithm. Indeed, Table 1 summarizes a computational experiment which supports this statement. In the table, we present the results of using the standard CG to solve 272 instances of the one-dimensional cutting stock problem (CSP), 87 instances of the vehicle routing problem with time windows (VRPTW), and 751 instances of the capacitated lot sizing problem with setup times (CLSPST). For details regarding these instances and problems, see [12]. Two versions of the CG method were implemented using the IBM CPLEX linear programming solvers. In both versions the RMPs are solved to optimality, but using one of the following two methods: the primal simplex method and the interior point method (barrier) without crossover (results of the dual simplex method are not included as this method was inferior to the primal in the CG context). For each class of problems, Table 1 shows for the two versions: the total number of iterations (Iter), the total CPU time spent solving the pricing subproblems (Pricing) and the total CPU time to solve the instances (Total), both in seconds, for all the instances in each class. The results support the point that using the interior point algorithm for obtaining the *optimal* solutions of the RMPs typically does not improve the performance of the CG technique. However, it does not mean that the interior point method is useless in the CG context. Indeed, when appropriately exploited, this algorithm offers advantageous features which can significantly improve the column CG method. One of the first advantages is that interior point methods can effectively stabilize CG. Indeed, when the simplex method is used to optimize the RMP, the dual optimal solutions correspond to extreme points of the dual feasible set, which has a negative impact on the performance of CG. In fact, extreme optimal solutions oscillate too much between two consecutive iterations. This behavior leads to a slow convergence of the CG technique, specially in the last iterations. To overcome this, we may use non-optimal dual solutions instead. In fact, optimal solutions are necessary only at the last iteration of the CG technique (to prove optimality).

	CG with simplex (primal)			CG with interior point (barrier)		
	Iter	Pricing (s)	Total (s)	Iter	Pricing (s)	Total (s)
CSP	18252	5173.05	5667.29	13301	4246.12	5311.55
VRPTW	2833	1303.45	1313.07	2370	1258.23	1339.46
CLSPST	33557	2622.69	2647.42	32043	2713.99	3282.01

Table 1: Results of the standard CG using optimal solutions obtained by the simplex method and the interior point method.

Many different strategies have been proposed in the literature with the purpose of stabilizing CG using non-optimal solutions [24, 6, 16, 10, 9, 22, 2, 1, 12]. In most of them, artificial elements (such as bounds, variables and penalties) are added to the RMP, so that the dual solutions stay close to a previously computed dual solution. Hence, these approaches require changing the formulation of the RMP and, in some of them, the changes may increase the difficulty of solving the problem. In addition, the stability of the dual solutions depends strongly on the choice of several parameters for which the best setting typically vary from problem to problem. On the

other hand, approaches relying on the interior point method [10, 12] have two main features: (i) they do not require the RMP to be modified; (ii) they rely on central prices, i.e., dual solutions which are well-centered in the dual feasible set. As a result, they naturally reduce the oscillation of dual solutions in consecutive iterations of the CG algorithm. To illustrate this behavior, Fig. 1 presents the results of solving the VRPTW instance `C107` [23] by using two CG variants: one is the standard CG (optimal dual solutions obtained with the simplex method), and the other is the primal-dual column generation [12] (non-optimal dual solutions obtained with the primal-dual interior point method). In the figure, we plot the distances $\|u^i - u^{i+1}\|_2$ between two consecutive dual solutions u^i and u^{i+1}, for each iteration i of the variants (the solution vectors were normalized). As the plot suggests, non-optimal dual solutions provided by the interior point method oscillate less than extreme optimal dual solutions.

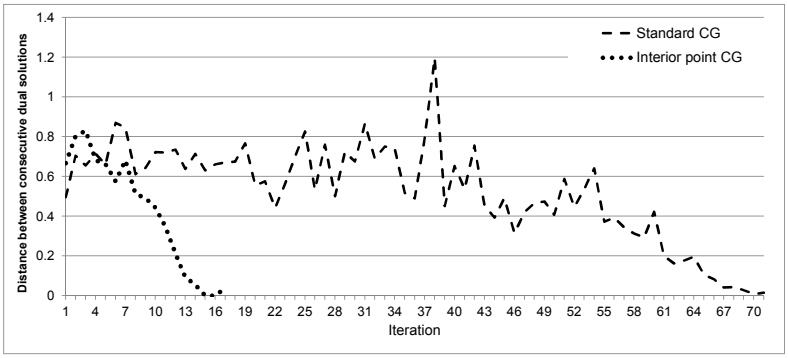

Figure 1: Oscillation of consecutive dual solutions in the standard CG and in a stabilized CG.

The primal-dual column generation method (PDCGM) takes advantage of the centrality of the dual solutions provided by the primal-dual interior point method [12]. Instead of solving each RMP to optimality, the PDCGM use suboptimal dual solutions which are also well-centered in the feasible set, as the iterates of the primal-dual interior point method stay close to the central path of the feasible set [11]. Given a primal-dual feasible solution $(\tilde{\lambda}, \tilde{u})$ of the RMP, we call it *suboptimal* if it satisfies $0 < (c^T\tilde{\lambda} - b^T\tilde{u}) \leq \varepsilon(1 + |c^T\tilde{\lambda}|)$, for a given parameter $\varepsilon > 0$, which measures the distance to optimality. The parameter ε is dynamically adjusted according to the relative gap. At each iteration, it is given by $\varepsilon = \min\{\varepsilon_{\max},\ gap/D\}$, where $\varepsilon_{\max} > 0$ is a fixed upper bound (*e.g.*, $\varepsilon_{\max} = 1.0$) and $D > 1.0$ is the degree of optimality which controls the reduction of ε. The relative gap is computed as

$$gap = \frac{c^T\tilde{\lambda} - (\kappa z_{SP}(\tilde{u}) + b^T\tilde{u})}{1 + |c^T\tilde{\lambda}|},$$

where $\kappa > 0$ is an upper bound for $\sum_{j \in N} \lambda_j^*$ and λ^* is an optimal solution of the master problem (this value is usually known for a given problem). Therefore, in the first iterations of the PDCGM, the suboptimal solutions of the RMPs may have a large distance to optimality, as the relative gap is typically large at this stage. This is reasonable as the RMP is still a rough approximation of the MP in the early iterations. However, as the relative gap starts to reduce, the RMP becomes more accurate and thus it makes sense to get suboptimal solutions which are closer to optimality. At the last iteration, the relative gap is already very small, so the suboptimal solution will be sufficiently close to optimality. As proved in [12], the PDCGM converges to an optimal solution of the MP, even though suboptimal solutions are used in the intermediate iterations.

To illustrate the impact of using well-centered and suboptimal solutions we have run computational experiments using the PDCGM for solving the same classes of problems presented in Table 1. Table 2 reports the results obtained with the PDCGM using suboptimal dual solutions (default setting) and optimal dual solutions (i.e., at each iteration, the RMP is solved to

optimality by the interior point method). We observe that the total number of iterations was smaller for the PDCGM with optimal solutions, regarding CSP and VRPTW. However, since going for optimality at each iteration required additional CPU time, the PDCGM with suboptimal solutions resulted in the best total CPU time for the three classes of problems. Notice that using optimal solutions adversely affects the pricing subproblems as well. Therefore, the results indicate that relying on suboptimal solutions is an advantageous feature of PDCGM. In addition, the total number of iterations and total CPU times were smaller than those obtained with the standard CG, as we can observe in Table 1. This comparison demonstrates the benefits of using well-centered solutions, as the results in Table 1 rely on extreme optimal solutions obtained with the simplex method.

	Subptimal dual solutions			Optimal dual solutions		
	Iter	Pricing (s)	Total (s)	Iter	Pricing (s)	Total (s)
CSP	16499	2948.56	3945.48	12911	3436.38	4967.76
VRPTW	914	470.38	545.50	684	723.84	910.45
CLSPST	18007	1132.36	1448.73	18036	1130.26	1798.78

Table 2: Results of using optimal and suboptimal dual solutions in the primal-dual column generation method (PDCGM).

3 Branch-and-Price with the Interior Point Method

The branch-and-price (BP) consists in using the CG technique within the branch-and-bound method in order to solve integer programming formulations with a huge number of variables [15, 4]. We can use the interior point algorithm to improve the performance of the BP method as well, following similar ideas as those in the previous section. However, the BP includes not only column generation, but also branching strategies, heuristics and, possibly, the generation of valid inequalities (branch-price-and-cut). Hence, a natural question would be: Can we take benefit of using the well-centered suboptimal solutions of the interior point method within these other operations as well? Although only a few papers in the literature have addressed such combination [19, 7, 21], the answer to this question is "*yes*".

Branching is typically based on optimal solutions. Hence, at each node of the BP, we apply the CG method to find the optimal solution of the master problem (MP) and, in case this solution is fractional, we use a branching strategy to decide how to branch at the current node. However, a branching strategy is actually a way of guessing what the best branching decision at the moment is. None of them guarantees a decision which leads to the best overall performance for solving an arbitrary problem. Hence, even if we branch using the optimal solution of the master problem, we cannot guarantee such branching decision is the best. As a result, going for optimality may be a waste of time, if we could anticipate which variables will be fractional at the optimal solution. This is the idea behind the *early branching* strategy. Instead of solving the MP to optimality, we terminate the CG algorithm prematurely, by setting a loose optimality tolerance (e.g., $\delta = 10^{-3}$). The obtained solution may not be an optimal one, but it should be a good approximation of it. It is typically easier to satisfy such requirement if we use well-centered solutions instead of extreme optimal solutions. Indeed, this is supported by the computational experiments presented in [21] for a branch-price-and-cut method which uses the primal-dual interior point method. Well-centered and suboptimal solutions are used to generate columns and valid inequalities and, in particular, to perform early branching. Table 3 summarizes the importance of early branching on this interior point branch-price-and-cut method, using the results of 29 VRPTW instances with 100 customers each (from 100-series of Solomon [23]). Four different early branching strategies have been tested, each one using a different value for the optimality tolerance δ. For each strategy, Table 3 gives the total number of nodes and valid inequalities (cuts) and the total CPU time, in seconds, to solve the 29 instances. As the results

DOI: 10.5540/03.2015.003.01.0525

indicate, early branching with $\delta = 10^{-3}$ was an effective technique for reducing all the considered measures, in particular the total CPU time.

Tolerance (δ)	Nodes	Cuts	CPU time (s)
10^{-3}	269	3331	8120.90
10^{-4}	277	3165	8846.00
10^{-5}	317	3581	10907.46
10^{-6}	303	3408	15399.37

Table 3: Total number of nodes and valid inequalities (cuts) and total CPU time for solving 29 instances of the VRPTW using an interior point branch-price-and-cut method with different values of δ (the optimality tolerance used for early branching).

A similar idea can be used to improve the performance of separation procedures that generate valid inequalities. Namely, instead of solving the master problem to optimality and then generate cuts, we call the separation procedures in the course of the CG algorithm. More specifically, we merge the column and cut generation algorithms into one, so that at each iteration of the resulting algorithm we may call one of the two types of subproblem: (i) a pricing subproblem to generate new columns; or (ii) a separation subproblem to generate new valid inequalities. This way, the well-centered and suboptimal solutions provided by the primal-dual interior point method may be used to generate valid inequalities as well.

Table 4 collects the computational results of solving six challenging instances of the VRPTW using two different variants of the branch-price-and-cut method, as reported in [3, 21]. The first is the standard variant in which the simplex method is used to obtain optimal solutions which are used in the generation of columns and valid inequalities as well as in the branching operations. The second variant uses the primal-dual interior point method as described above. The results indicate that the interior point variant is consistently more efficient than the other approach.

	Standard (simplex) BPC			Interior point BPC		
	Nodes	Cuts	Time (s)	Nodes	Cuts	Time (s)
R108	1	296	891	1	153	289
R110	5	219	426	3	111	50
R112	19	574	16073	15	365	1768
RC207	5	210	91405	3	114	15847
R206	1	171	60608	1	57	2033
R210	5	266	400904	5	147	5234

Table 4: Computational results of solving six instances of the VRPTW with a standard branch-price-and-cut method and an interior point branch-price-and-cut method.

4 Concluding Remarks

In this paper, we described ways to improve the performance of the column generation technique and the branch-and-price method by using advantageous features offered by the primal-dual interior point algorithm. We presented numerical evidence to support that well-centered and suboptimal solutions may benefit the generation of columns and valid inequalities as well as improve branching. This way, we have summarized the main developments proposed in the thesis *Theoretical and computational issues for improving the performance of linear optimization methods* [20] which has been awarded with the *Doctoral Prize Odelar Leite Linhares* in 2014.

References

[1] H. M. Ben Amor, J. Desrosiers, and A. Frangioni. On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167 – 1184, 2009.

[2] O. Briant, C. Lemarechal, P. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical programming*, 113(2):299–344, 2008.

[3] G. Desaulniers, F. Lessard, and A. Hadjar. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404, 2008.

[4] J. Desrosiers and M. E. Lübbecke. *Branch-Price-and-Cut Algorithms*. John Wiley & Sons, Inc., 2010.

[5] O. du Merle, P. Hansen, B. Jaumard, and N. Mladenovic. An interior point algorithm for minimum sum-of-squares clustering. *SIAM J. Sci. Comput.*, 21(4):1485–1505, 1999.

[6] O. Du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1):229–237, 1999.

[7] S. Elhedhli and J.-L. Goffin. The integration of an interior-point cutting plane method within a branch-and-price algorithm. *Mathematical programming*, 100:267–294, 2004.

[8] L. R. Ford and D. R. Fulkerson. A suggested computation for maximal multi-commodity network flows. *Management Science*, 5(1):97–101, 1958.

[9] A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13:117–156, May 2002.

[10] J. L. Goffin and J. P. Vial. Convex nondifferentiable optimization: a survey focussed on the analytic center cutting plane method. *Optimization methods and software*, 17(5):805–867, 2002.

[11] J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

[12] J. Gondzio, P. Gonzalez-Brevis, and P. Munari. New developments in the primal-dual column generation technique. *European Journal of Operational Research*, 224(1):41–51, 2013.

[13] J. Gondzio and R. Sarkissian. Column generation with a primal-dual method. Technical report, Logilab, 1996.

[14] J. Kelley, J. E. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[15] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[16] R. K. Martinson and J. Tind. An interior point method in Dantzig-Wolfe decomposition. *Computers and Operation Research*, 26:1195–1216, 1999.

[17] J. Mitchell and B. Borchers. Solving real-world linear ordering problems using a primal-dual interior point cutting plane method. *Annals of Operations Research*, 62:253–276, 1996.

[18] J. E. Mitchell. Computational experience with an interior point cutting plane algorithm. *SIAM Journal of Optimization*, 10(4):1212–1227, 2000.

[19] J. E. Mitchell and M. J. Todd. Solving combinatorial optimization problems using Karmarkar's algorithm. *Mathematical Programming*, 56:245–284, 1992.

[20] P. Munari. *Theoretical and computational issues for improving the performance of linear optimization methods*. PhD thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2012.

[21] P. Munari and J. Gondzio. Using the primal-dual interior point algorithm within the branch-price-and-cut method. *Computers & Operations Research*, 40(8):2026 – 2036, 2013.

[22] L.-M. Rousseau, M. Gendreau, and D. Feillet. Interior point stabilization for column generation. *Operations Research Letters*, 35(5):660–668, 2007.

[23] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

[24] P. Wentges. Weighted Dantzig–Wolfe Decomposition for Linear Mixed-integer Programming. *International Transactions in Operational Research*, 4(2):151–162, 1997.

