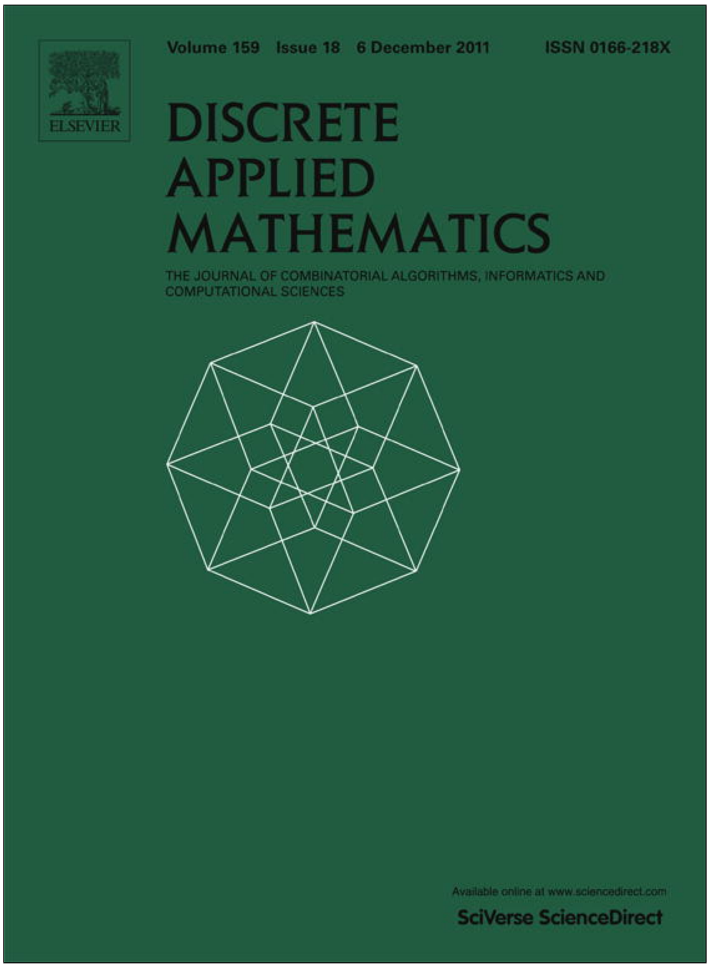

Volume 159 Issue 18 6 December 2011
ISSN 0166-218X
ELSEVIER
DISCRETE
APPLIED
MATHEMATICS
THE JOURNAL OF COMBINATORIAL ALGORITHMS, INFORMATICS AND COMPUTATIONAL SCIENCES
Available online at www.sciencedirect.com
SciVerse ScienceDirect

Discrete Applied Mathematics 159 (2011) 2251–2265

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

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

Chebyshev center based column generation

Chungmok Lee [a], Sungsoo Park [b,*]

[a] *IT Convergence Technology Research Laboratory, ETRI, 218 Gajeongno, Yuseong-gu, Daejeon, 305-700, Republic of Korea*

[b] *Department of Industrial and System Engineering, KAIST, 373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea*

ARTICLE INFO

Article history:
Received 19 August 2009
Received in revised form 1 August 2011
Accepted 5 August 2011
Available online 23 September 2011

Keywords:
Stabilized column generation
Chebyshev center
Binpacking problem
Vehicle routing problem
Generalized assignment problem

ABSTRACT

The classical column generation approach often shows a very slow convergence. Many different acceleration techniques have been proposed recently to improve the convergence. Here, we briefly survey these methods and propose a novel algorithm based on the Chebyshev center of the dual polyhedron. The Chebyshev center can be obtained by solving a linear program; consequently, the proposed method can be applied with small modifications on the classical column generation procedure. We also show that the performance of our algorithm can be enhanced by introducing proximity parameters which enable the position of the Chebyshev center to be adjusted. Numerical experiments are conducted on the binpacking, vehicle routing problem with time windows, and the generalized assignment problem. The computational results of these experiments demonstrate the effectiveness of our proposed method.

1. Introduction

The Dantzig–Wolfe decomposition algorithm, first introduced by Dantzig and Wolfe in 1961 [11], is considered to be one of the most successful methods for tackling large-size mathematical programming problems, especially integer programming problems [23]. The incorporation of the column generation method into the linear programming-based branch-and-bound scheme, often referred to as the branch-and-price scheme, is able to solve many real-life hard optimization problems [2,12,31].

The column generation procedure, which is based on the simplex algorithm, has the major drawback that it often shows desperately slow convergence. Vanderbeck [33] summarized several limitations of the simplex-based column generation procedure as follows: (i) slow convergence (the *tailing-off effect*); (ii) poor columns in the initial stage (the *head-in effect*); (iii) the optimal value of the restricted master problem remains the same during many iterations (the *plateau effect*); (iv) the dual solution jumps from one extreme point to another (the *bang–bang effect*); (v) the intermediate Lagrangian dual bounds do not converge monotonically (the *yo-yo effect*). In column generation, the main driving force is the dual solution of the restricted master problem. One of the major sources of troubles is the instability of the dual solutions that can arise from the dual degeneracy and the extremity of the dual optimal solutions when the simplex algorithm is used. The lack of good columns in the current restricted master problem also contributes to the instability of this method. When viewed as a classical convex optimization approach, column generation can be understood as an iterative method of updating Lagrangian multipliers (the dual variables) until the optimality condition (the complementary slackness condition) is satisfied, using the column generation subproblem and the dual solution of the linear programming problem (the master problem). More specifically, classical column generation essentially involves the same process as applying the Kelley's cutting plane algorithm [22] to the dual of the restricted master problem. Note that the dual problem has a restricted set of inequalities. In Kelley's method, the maximally violated inequality at the current solution is obtained via the *oracle*

* Corresponding author. Tel.: +82 42 350 3121; fax: +82 42 350 3110.
E-mail addresses: chungmok@gmail.com (C. Lee), sspark@kaist.ac.kr (S. Park).

0166-218X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2011.08.009

(the separation problem) and then added to the problem. The maximally violated inequality corresponds to the column with the smallest (in a minimizing problem) reduced cost in the primal problem, and the column generation subproblem functions as the oracle. However, Kelley's method can be desperately slow [36], so it is not surprising that classical column generation may also be desperately slow. Many acceleration schemes for column generation have recently been proposed in the literature [4–6,16,29,30,28,34,35]. The underlying rationale for most of them is stabilization of the dual solutions by *penalizing* the distance between the new dual solution and the best dual solution obtained so far; these approaches are often referred to as the *stabilized column generation* scheme. Since column generation can be understood as a cutting plane method in the dual perspective, any acceleration scheme for the cutting planes can be adopted within the framework of column generation. Stated simply, the stabilized column generation approach can be understood as a special case of the *bundle method* that is commonly used in the convex optimization field. Another well-known approach for accelerating cutting planes is achieved by *centering* the new dual solution within the dual polyhedron. Among the centering methods that have been developed, much attention has been paid to the analytic center cutting plane method (ACCPM) [20,26] as an approach for solving the convex optimization problems. Elzinga and Moore [18] have proposed a cutting plane method based on the Chebyshev center, which is another well-known center of a convex set. The fact that the Chebyshev center can be obtained by solving a simple linear program is a major advantage that allows researchers to keep using the linear programming based branch-and-price scheme. Despite the existence of a theoretical link between the cutting plane method and the column generation method, to the best of our knowledge, there has as yet been no attempt to develop a method using the Chebyshev center within the column generation scheme for solving integer programming problems. In this paper, we develop and investigate the effectiveness of the Chebyshev center based column generation on several well-known integer programming problems. Moreover, we propose a novel acceleration scheme for the Chebyshev center algorithm, in which the distance between the best dual bound cut and the next Chebyshev center is adjusted adaptively during column generation.

In Section 2, we give a brief overview of the various acceleration methods used for column generation. In Section 3, the Chebyshev center based column generation scheme is proposed, and numerical results of the proposed algorithm and comparisons with other methods are reported in Section 4. In Section 5, we summarize our data and present our conclusions.

2. Accelerating column generation procedure

Let X be a finite set of vectors $x \in R^n$, $X^k \subseteq X$ and K be an index set of the elements of X^k, i.e., $\{x^i | i \in K\} = X^k \subseteq X$. Here, X may represent a set of the integer vectors satisfying some linear inequalities or a set of extreme points of a polytope. Let A denote an $m \times n$ real valued matrix and $b \in R^m$, $c \in R^n$ denote real valued column vectors. Then, consider an optimization problem which is given as

$$\begin{aligned} \min \quad & c^T x && (1)\\ \text{subject to} \quad & Ax \geq b, && (2)\\ & x \in conv(X), && (3) \end{aligned}$$

where $conv(X)$ is the convex hull of X. If only a restricted set $X^k \subseteq X$ is known, we have a restricted version of the above optimization problem as follows:

$$\begin{aligned} \min \quad & c^T x && (4)\\ \text{subject to} \quad & Ax \geq b, && (5)\\ & x \in conv(X^k). && (6) \end{aligned}$$

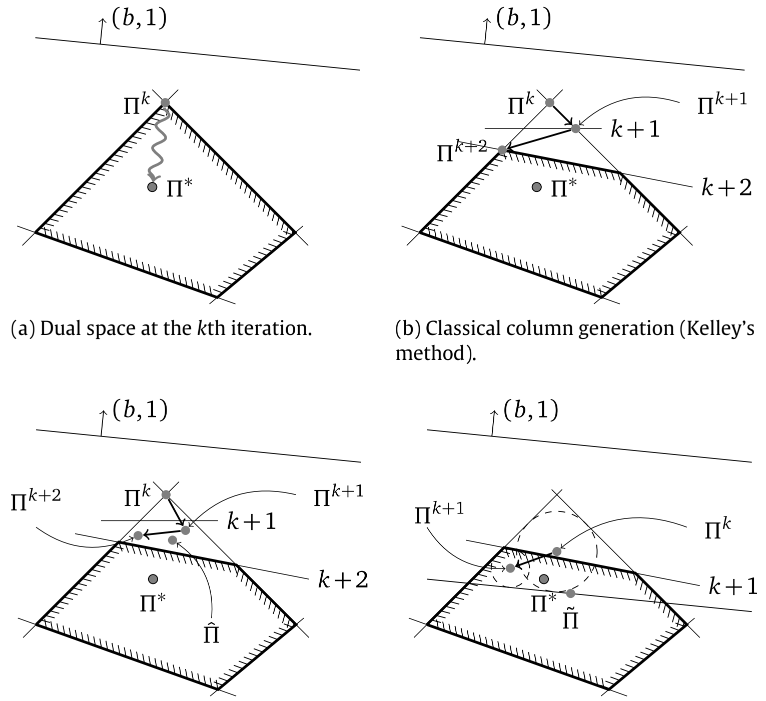
By the Dantzig–Wolfe decomposition method, the above problem can be reformulated as follows, which we call the *restricted master problem*:

$$\begin{aligned} \min \quad & \sum_{i \in K} c^T x^i \lambda_i && (7)\\ \text{subject to} \quad & \sum_{i \in K} A x^i \lambda_i \geq b, && (8)\\ & \sum_{i \in K} \lambda_i = 1, && (9)\\ & \lambda_i \geq 0, \quad \forall i \in K. && (10) \end{aligned}$$

And its dual:

$$\begin{aligned} \max \quad & f_D(\pi, \pi_0) = b^T \pi + \pi_0 && (11)\\ \text{subject to} \quad & (Ax^i)^T \pi + \pi_0 \leq c^T x^i, \quad \forall i \in K, && (12)\\ & \pi \geq 0, && (13) \end{aligned}$$

where $\pi \in R^m$ and $\pi_0 \in R$ are the dual vectors associated with constraints (8) and (9), respectively.



(a) Dual space at the kth iteration.

(b) Classical column generation (Kelley's method).

(c) Stabilized column generation.

(d) Column generation based on the Chebyshev center.

Fig. 1. Π^* is the true dual optimal solution. Π^k is the optimal solution of the current relaxed dual problem.

Note that the dual of the restricted master problem has a restricted set of constraints, making it a *relaxed* dual problem. Let $\Pi^* = (\pi^*, \pi_0^*)$ be the optimal solution of the unrelaxed dual problem (all inequalities of which are obtained from the unrestricted set of feasible $x \in X$), and $\Pi^k = (\pi^k, \pi_0^k)$ denote the dual optimal solution of the dual problem with all inequalities $i \in K$ or, equivalently, the primal problem with all columns $i \in K$. Let $\tilde{\Pi} = (\tilde{\pi}, \tilde{\pi}_0)$ be a feasible dual solution to the unrelaxed dual problem, i.e., the column generation oracle cannot find any negative reduced cost column for the dual vector $\tilde{\Pi} = (\tilde{\pi}, \tilde{\pi}_0)$. Clearly $f_D(\tilde{\pi}, \tilde{\pi}_0) \leq f_D(\pi^*, \pi_0^*) \leq f_D(\pi^k, \pi_0^k)$ holds. The dual space at iteration k of the column generation procedure is illustrated in Fig. 1(a). The dual feasible set is relaxed, so the true optimal dual solution may be inside of the feasible set. Observe that during the column generation iterations, the dual solutions are *enhanced* toward (π^*, π_0^*). Therefore, it is crucial that one should choose an adequate next dual vector (π^{k+1}, π_0^{k+1}). In the following subsections, we briefly review several dual updating methods, and propose a new dual updating scheme based on the Chebyshev center.

2.1. Kelley's cutting plane method

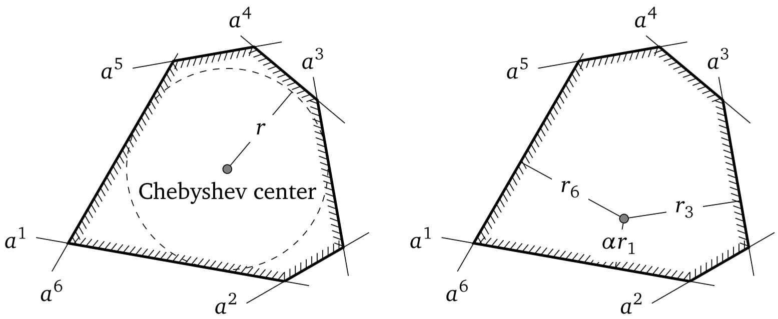
Kelley's method is equivalent to the dual interpretation of the classical column generation method based on the simplex algorithm [22], as illustrated in Fig. 1(b). As shown in the figure, the dual solution is likely to move among extreme points of the dual polyhedron. When there are many constraints whose defining coefficient vectors are nearly parallel to the cost vector $(b, 1)$, the zigzag movement of the dual solution may be apparent.

2.2. Wentges' weighted Dantzig–Wolfe decomposition

In 1997, Wentges introduced the weighted Dantzig–Wolfe decomposition method for linear mixed-integer programming [34]. The subsequent Lagrangian multiplier (dual solution) was obtained by a convex combination of the current solution and the solution with the best Lagrangian bound obtained to date. The underlying motivation of this approach was to search a better dual solution in the neighborhood of the best solution. This method can be interpreted as a direction search procedure that always starts from the solution with the best Lagrangian bound, say Π^{best}, with direction Π^k–Π^{best}. Note that Π^{best} is not feasible for the unrelaxed dual problem, so the procedure is terminated when Π^{best} is unrelaxed dual feasible. In his paper, Wentges reported the improved computational results of the weighted Dantzig–Wolfe decomposition method on the capacitated facility location problem especially in terms of the number of iterations, compared with the classical column generation method.

2.3. Stabilized column generation

The stabilized column generation approach can be seen as a special case of the *bundle method*, which is well known method in the convex optimization field. In the bundle method, the dual solution is often *constrained* to a given interval, and



(a) Chebyshev center of a polyhedron over Euclidian norm.

(b) Proximity adjusted Chebyshev center of a polyhedron over Euclidian norm. The polyhedron is defined by six inequalities, and the Chebyshev center is the point that maximizes r, s.t. $a^1x + \alpha\|a^1\|r \leq b^1$, $a^ix + \|a^i\|r \leq b^i$, $i = 2, \ldots, 6$. Note that, at the proximity adjusted Chebyshev center, $r_1 = r_3 = r_6$ holds. The distance from the Chebyshev center to inequality a^1 is αr_1. The center point is getting closer to a^1 when $\alpha < 1$.

Fig. 2. Chebyshev center and the proximity-adjusted Chebyshev center.

any deviation from the interval is penalized by a penalty function. In this method, various penalty functions can be used, including boxstep [24], polyhedral penalty [16,27,29,35] and quadratic function [5,15]. The boxstep and polyhedral penalty can be applied via reformulation of the original linear programming problem, while the nonlinear programming model is necessary for the quadratic penalty function. Although the quadratic convex optimization problem can be solved efficiently by the interior point method, one may lose many of the advantages of the linear programming, such as the applicability of very powerful linear programming solvers, when this method is used. The stabilized column generation method can be understood as a bundle method applied on the dual of the restricted master problem with a polyhedral penalty function. The penalty function for the stabilized column generation is often defined as a simple V-shaped function whose center is specified as the *stabilizing center*, and slope ϵ determines how much the distance from the best dual solution is penalized. A large number of reports have been published in recent years on how to accelerate the column generation procedure, most of which are concerned with these stabilization techniques (see [5,9,23,16,27,29,35]).

To implement stabilized column generation, the stabilization center and the penalty parameter ϵ must be determined. Moreover, in order to ensure optimality of the original problem, the stabilization center and the penalty parameter ϵ should be adjusted carefully throughout the algorithm. It would appear that the performance of the stabilization method depends largely on how to manage these parameters. In specific contexts, problem specific knowledge may help improve the performance. For example, the initial stabilization center can be constructed from the relaxed original formulation. Oukil et al. considered the multiple-depot vehicle scheduling problem (MDVSP) [27]. They solved the single depot version of MDVSP, which is the relaxation of the MDVSP, and used its dual solution as the stabilization center for the MDVSP. The experimental results show that this good initial stabilization center can significantly accelerate column generation. The stabilization technique has been proven to be very effective in accelerating column generation for various problems [5,16,27]. For a detailed comparison of the bundle method and the classical column generation approach the reader is referred to the survey paper by Briant et al. [9]. The behavior of dual solutions during the stabilized column generation is illustrated in Fig. 1(c).

3. Column generation based on the Chebyshev center

For a bounded, closed, and nonempty convex set, the Chebyshev center is the deepest point inside the set, in the sense that it is farthest from the exterior [8]. The Chebyshev center of a polyhedron defined by some linear inequalities is shown in Fig. 2(a). If the convex set is defined by a set of linear inequalities $a_i^T x \leq b_i$, $\forall i \in \{1, \ldots, m\}$, the Chebyshev center can be found by solving the linear program [8]:

$$\max \quad r \tag{14}$$

$$\text{subject to} \quad a_i^T x + \|a_i\|_* r \leq b_i, \quad \forall i \in \{1, \ldots, m\}, \tag{15}$$

$$r \geq 0, \tag{16}$$

where $\|a\|_*$ is any norm of vector a.

Elzinga and Moore developed the cutting plane algorithm using the Chebyshev center in a rather general sense, i.e., without a specific target problem [18]; in the same report, they also described the convergence property of the Chebyshev center based cutting plane method. Betrò proposed an accelerated cutting plane scheme using the Chebyshev center by introducing a *deeper objective cut*, which is obtained by perturbing the dual feasible solution [7]. The Chebyshev center cutting plane method is a cutting plane method based on the concept of *centering*. An important variant of the classical cutting plane method is the ACCPM in which the analytic center is considered to be the point that maximizes the product of the slacks of given inequalities. The ACCPM and bundle method can be combined to improve convergence [1,15], and some authors have used the ACCPM in column generation scheme; for example, Elhedhil and Goffin solved the binpacking problem using the ACCPM [17], and Goffin et al. examined the multicommodity network flow problem [19]. However, to the best of our knowledge, no study has been devoted to evaluating the Chebyshev center based method with the column generation procedure.

Here, we adopt the Chebyshev center cutting plane method to the column generation scheme via the dual perspective. Consider the dual of the restricted master problem: max (11), s.t. (12) and (13). We assume that the dual polyhedron is bounded and has an interior point (if not, we consider a subset of inequalities of the dual problem which has an interior point). If a dual bound $\tilde{Z}$ or a dual feasible solution $\tilde{\Pi}$ (we set $\tilde{Z}$ as $b\tilde{\pi} + \tilde{\pi}_0$) is known, a trivial inequality $-b\pi - \pi_0 \le -\tilde{Z}$ can be added to the dual problem. We reformulate the dual problem of the restricted master problem as follows, which we call the *Chebyshev dual master problem*:

$$\max \quad r \tag{17}$$

$$\text{subject to} \quad (Ax^i)^T\pi + \pi_0 + \|(Ax^i, 1)\|_* r \le c^T x^i, \quad \forall i \in K, \tag{18}$$

$$-\pi_j + r \le 0, \quad \forall j = 1, \dots, m, \tag{19}$$

$$-b^T\pi - \pi_0 + \|(b, 1)\|_* r \le -\tilde{Z}, \tag{20}$$

$$\pi \ge 0, \tag{21}$$

$$r \ge 0. \tag{22}$$

The above problem is to find the Chebyshev center point defined by the restricted set of inequalities K, and the dual bound. Let $y \in R^m$ and $z \in R$ denote the primal variables associated with inequalities (19) and (20), respectively. Then, the restricted master problem can be given as follows, which we call the *Chebyshev primal master problem*:

$$\min \quad \sum_{i \in K} c^T x^i \lambda_i - \tilde{Z} z \tag{23}$$

$$\text{subject to} \quad \sum_{i \in K} Ax^i \lambda_i - y - b^T z \ge 0, \tag{24}$$

$$\sum_{i \in K} \lambda_i - z = 0, \tag{25}$$

$$\sum_{i \in K} \|(Ax^i, 1)\|_* \lambda_i + \sum_{j=1}^{m} y_j + \|(b, 1)\|_* z \ge 1, \tag{26}$$

$$\lambda_i \ge 0, \quad \forall i \in K, \tag{27}$$

$$y_j \ge 0, \quad \forall j = 1, \dots, m, \tag{28}$$

$$z \ge 0. \tag{29}$$

The distance from a hyperplane to the Chebyshev center is defined by the norm. In particular, for the Euclidean norm (L_2 norm), $\|a\|$ is $\sqrt{\sum_{i \in N} |a_i|^2}$, where $a \in R^N$.

Note that the Chebyshev primal master problem has more variables $y \in R^m$, $z \in R$ and one more constraint (26) than the original restricted master problem. The column generation procedure, however, can be performed in almost the same manner as the original restricted master problem. Suppose that we solved the Chebyshev primal master problem and obtained the dual optimal solution (π', π'_0), where π' and π'_0 are optimal dual variables associated with constraints (24) and (25), respectively. To find an entering column, we solve the pricing subproblem: min $(c^T - \pi'^T A)x - \pi'_0$, s.t. $x \in X$. Let z' and x' be the optimal value and an optimal solution to the pricing subproblem. If $z' < 0$, (π', π'_0) is violated by the constraint (18) identified with x', hence the new column $(Ax', 1, \|(Ax', 1)\|)$ can be added to the Chebyshev primal master problem. Note that $\|(Ax', 1)\|_*$ can be calculated easily once Ax' is identified. If $z' \ge 0$, (π', π'_0) is a feasible solution to the unrelaxed dual problem and consequently, the best dual bound $\tilde{Z}$ can be updated to $(b, 1)^T(\pi', \pi'_0) = b^T\pi' + \pi'_0$. Note that we do not need to consider the dual variable r for constraint (26) for pricing because $z' - \|(Ax^i, 1)\|_* r \le z'$, and we do not skip any violated cut (or beneficial column). Therefore, the column generation subproblem is the same as the subproblem for the original restricted master problem. This procedure is repeated until the optimal value of the above problem becomes zero. An optimal value of zero for the above problem implies that the radius of the Chebyshev sphere is also zero from the strong duality of the primal and the dual linear programs. In the practical implementation, some positive value ε can be used instead of zero in the terminating condition. It is noteworthy that the use of a fixed value of ε may not be a good strategy because in the Chebyshev master problem, the value of the original primal objective function $\sum_{i\in K} c^T x^i \lambda_i$ may be very small when $z \ll 1$ by (25), which leads to an unexpected early termination of the algorithm. One remedy is to use $\varepsilon z'$ as the termination criterion; $\sum_{i\in K} c^T x^i \lambda_i - \tilde{Z}z < \varepsilon z \Rightarrow \sum_{i\in K} c^T x^i \tilde{\lambda}_i - \tilde{Z} < \varepsilon$, where $\tilde{\lambda} := \lambda/z$ and $\sum_{i\in K} \tilde{\lambda}_i = 1$ by (25). The detailed algorithm of the column generation based on the Chebyshev center is presented in Algorithm 1.

Algorithm 1 Column generation based on the Chebyshev center

procedure CHEBYSHEVCENTERCOLGEN
  $\tilde{Z} \leftarrow 0$ ▷ Or any valid dual bound
  repeat
    Solve the restricted Chebyshev primal master problem
    (λ', y', z') and (π', π_0') ← the optimal primal and dual solutions
    if the optimal value $\sum_{i\in K} c^T x^i \lambda_i' - \tilde{Z}z'$ is greater than $\varepsilon z'$ **then** ▷ Equivalently $r > \varepsilon z'$
      Solve the column generation subproblem using (π', π_0') as dual solution
      if new column is identified with x' **then** ▷ Reduced cost is negative
        Add new column $(Ax', 1, \|(Ax', 1)\|)$ to the problem
      else
        $\tilde{Z} \leftarrow b^T \pi' + \pi_0'$ ▷ Update the best dual bound
      end if
    end if
  until $r \le \varepsilon z'$
end procedure

In the classical column generation approach, the master problem is solved to obtain the next dual solution which lies on a vertex point of the dual polyhedron. In the Chebyshev center based column generation, the next dual solution corresponds to the Chebyshev center of the polyhedron which is the intersection of the current relaxed dual polyhedron and the half space constrained by the best dual bound identified so far. In Fig. 1(d), $\tilde{\Pi}$ is the best known dual solution which is feasible for the *unrestricted* dual problem, and Π^k is the Chebyshev center point at iteration k. After the inequality $k + 1$ is added, the new Chebyshev center point Π^{k+1} is obtained. Note that the radius of the Chebyshev ball decreases or remains the same with the addition of inequalities.

3.1. Proximity adjusted Chebyshev center

When the Chebyshev dual master problem is being considered, the polyhedron of the problem is characterized by three types of inequalities: (i) inequalities of the generated columns $(Ax^1, 1, \|(Ax^1, 1)\|), \ldots, (Ax^k, 1, \|(Ax^k, 1)\|)$, referred to as *column inequalities*; (ii) the best dual bound, referred to as *best dual bound inequality*; (iii) inequalities for nonnegativity. Some of these inequalities may be more *important* than others. A large number of column inequalities and one best dual inequality together define the dual polyhedron at some iteration. In general, the column inequalities attempt to *push* the next Chebyshev center down (closer to the best dual inequality), while the best dual inequality tries to *lift* it up (closer to the column inequalities), as shown in Fig. 1(d). The relative position of the next Chebyshev center from an inequality can be adjusted by multiplying some value to the norm of a vector that defines the coefficient of r in the linear program for the Chebyshev center problem. We denote this as the proximity adjusted Chebyshev center, which is illustrated in Fig. 2(b).

We now apply the proximity parameter α to the best dual bound inequality (20). This can be done by multiplying some $\alpha > 0$ to the coefficient of r in (20). The proximity adjusted Chebyshev primal master problem can then be stated as follows:

min (23)

subject to (24), (25)

$$\sum_{i\in K} \|(Ax^i, 1)\|_* \lambda_i + \sum_{j=1}^{m} y_j + \alpha \|(b, 1)\|_* z \ge 1, \tag{30}$$

(27), (28), (29).

The value of α determines how far the next Chebyshev center point is located from the dual bound inequality. It is also possible that one modifies the value of α dynamically during the column generation procedure. As the number of columns is finite in a typical column generation subproblem for a combinatorial optimization problem, the number of column inequalities is also finite. Consequently, from now on, we assume that the number of columns is finite and that proximity parameter $\alpha_k > 0$ for any iteration k. We also assume that the dual polyhedron of the unrelaxed dual master problem is bounded and closed.

Theorem 3.1. *The optimal value of the proximity adjusted Chebyshev dual master problem, which is equivalent to the primal master problem, converges to zero — unless the sequence of positive numbers $\{\alpha_k\}$ converges to zero.*

Proof. Considering the dual polyhedron, it is easily seen that the sequence of radii of Chebyshev spheres $\{r_k\}$, is bounded and monotonically nonincreasing. Let S be the set of feasible solutions to the unrelaxed dual master problem. For Chebyshev center Π^k at iteration k, if $\Pi^k \notin S$ then the column generation subproblem eventually finds a cut which separates Π^k. Since there are finitely many columns, there exists n_0 such that $\Pi^l \in S$ for any $l > n_0$. Suppose that $\lim_{k\to\infty} r_k = \hat{r} > 0$. For any $l > n_0$, $(b, 1)^T \Pi^{l+1} \geq (b, 1)^T \Pi^l + \alpha_{l+1}\|(b, 1)\| r_{l+1}$ holds, because of (20). By the Cauchy–Schwarz inequality, we have $\|(b, 1)^T\| \|\Pi^{l+1} - \Pi^l\| \geq (b, 1)^T(\Pi^{l+1} - \Pi^l) \geq \alpha_{l+1}\|(b, 1)\| r_{l+1}$, which implies $0 < \hat{r} \leq r_{l+1} \leq \|\Pi^{l+1} - \Pi^l\|/\alpha_{l+1}$. If sequence $\{\alpha_k\}$ converges to $\hat{\alpha}$ such that $\hat{\alpha} > 0$, $\lim_{l\to\infty} \|\Pi^{l+1} - \Pi^l\| \geq \hat{r}\hat{\alpha} > 0$, which derives a contradiction, since S is bounded and closed. If $\{\alpha_k\}$ diverges but is bounded (e.g., oscillation between numbers), we have $\|\Pi^{l+1} - \Pi^l\| \geq \tilde{\alpha}\hat{r} > 0$ for any l, where $\tilde{\alpha} = \inf_{\forall l}\{\alpha_l\} > 0$, which again derives a contradiction, since S is bounded and closed. If $\{\alpha_k\}$ diverges towards infinity, clearly, $\{r_k\}$ should converge to zero, which contradicts the assumption. The sequence $\{r_k\}$, therefore, converges to zero. □

Since the proximity parameter α controls the proximity between the next position of Chebyshev center and the best dual inequality in the dual space, the column generation procedure may be accelerated if the value of α is properly chosen. Intuitively, the value of α should be small when we want to update the dual bound quickly. On the other hand, a large value of α may produce more relevant new columns, since the next Chebyshev center is becoming closer to the true optimal solution (extreme point) of the relaxed dual polyhedron. Based on these observations, we propose a heuristic algorithm to update the value of α. The principle of the algorithm is to increase the value of the proximity parameter α when updating of the dual bound fails. If the dual bound is updated, we reset the value of α to a small value in order to search for another quick update of the dual bound. Let T and Ω denote some given positive numbers. α is increased gradually up to Ω when the best dual value is not updated during the last T iterations. If the best dual is updated, we reset α to 1, and the procedure is repeated. See Algorithm 2 for details. Note that there are two parameters here, T controls how fast the value of α will increase, while Ω restricts the maximum value of α. Fig. 3(a) demonstrates the change in α for the proximity adjusted Chebyshev algorithm with $T = 10$ and $\Omega = 10$.

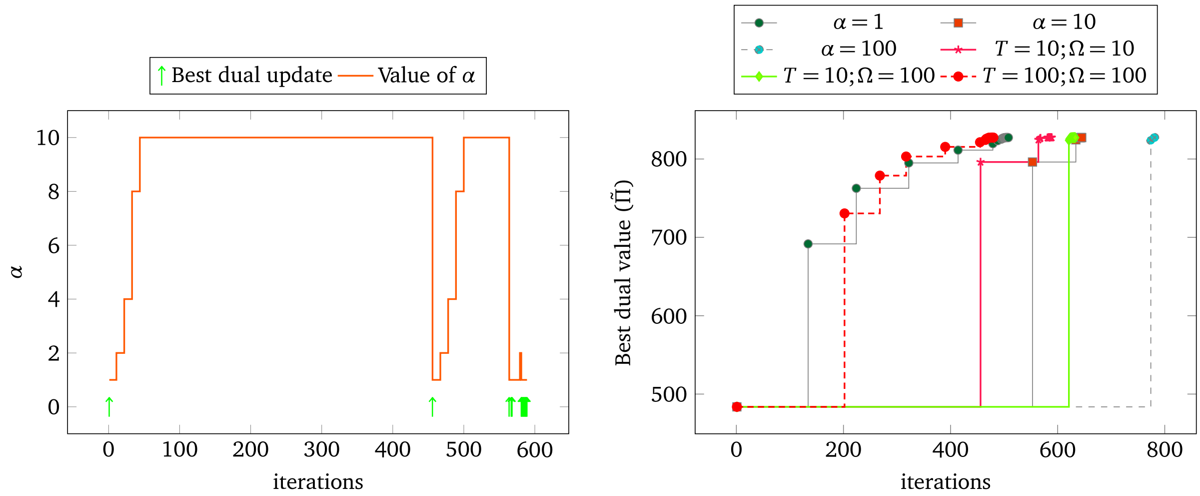
Algorithm 2 Column generation based on the Proximity Adjusted Chebyshev center

procedure PACHEBYSHEVCENTERCOLGEN
    $\tilde{Z} \leftarrow 0$ ▷ Or any valid dual bound
    $n \leftarrow 0$
    $\alpha \leftarrow 1$
    repeat
        Solve the restricted Chebyshev primal master problem
        (λ', y', z') and $(\pi', \pi'_0) \leftarrow$ the optimal primal and dual solutions
        if the optimal value $\sum_{i\in K} c^T x^i \lambda'_i - \tilde{Z}z'$ is greater than $\varepsilon z'$ **then** ▷ Equivalently $r > \varepsilon z'$
            Solve the column generation subproblem using (π', π'_0) as dual solution
            if new column is identified with x' **then** ▷ Reduced cost is negative
                Add new column $(Ax', 1, \|(Ax', 1)\|)$ to the problem
                $n \leftarrow n + 1$ ▷ Increase the non-dual-update iteration number.
            else
                $\tilde{Z} \leftarrow b^T\pi' + \pi'_0$ ▷ Update the best dual bound
                $n \leftarrow 0$
                $\alpha \leftarrow 1$ ▷ Reset value of α.
            end if
        end if
        if $n > T$ **then**
            $\alpha \leftarrow \min\{2\alpha, \Omega\}$ ▷ Increase α by two times at every T non-dual-update iteration.
            $n \leftarrow 0$
        end if
    until $r \leq \varepsilon z'$
end procedure

Fig. 3(b) shows a typical example of the changes that occur in the dual bound in the algorithm when various parameter are used values for the vehicle routing problem (VRP) C101 (the detailed description of this problem can be found in Section 4). It is clearly shown that small α may lead to frequent updates of the dual bound.

3.2. Stabilized Chebyshev center algorithm

The Chebyshev center method can be modified to use the stabilization technique. We call this combined algorithm the *stabilized Chebyshev algorithm*. In the stabilized Chebyshev algorithm, the Chebyshev center is stabilized by using a linear



(a) Change of α for the proximity adjusted Chebyshev algorithm with $T = 10$ and $\Omega = 10$.

(b) Updates of dual bound for the different algorithmic parameters.

Fig. 3. Behavior of the algorithm for VRP problem C101.

penalty function with the penalty parameter ϵ. For a given penalty parameter ϵ and a stabilization center $(\tilde{\Pi}, \tilde{\Pi}_0)$, the primal master problem of the stabilized Chebyshev algorithm is stated as follows:

$$\min \quad \sum_{i \in K} c^T x^i \lambda_i - \tilde{Z} z + \sum_{j=1,\dots,m} \tilde{\Pi}_j(\delta_j^+ - \delta_j^-) + \tilde{\Pi}_0(\delta_0^+ - \delta_0^-) \tag{31}$$

$$\text{subject to} \quad \sum_{i \in K} A x^i \lambda_i - y - b^T z + \delta^+ - \delta^- \geq 0, \tag{32}$$

$$\sum_{i \in K} \lambda_i - z + \delta_0^+ - \delta_0^- = 0, \tag{33}$$

$$\sum_{i \in K} \|(Ax^i, 1)\|_* \lambda_i + \sum_{j=1}^{m} y_j + \|(b, 1)\|_* z \geq 1, \tag{34}$$

$$\lambda_i \geq 0, \quad \forall i \in K, \tag{35}$$

$$y_j \geq 0, \quad \forall j = 1, \dots, m, \tag{36}$$

$$z \geq 0, \tag{37}$$

$$\delta_j^+ \leq \epsilon, \qquad \delta_j^- \leq \epsilon, \quad \forall j = 1, \dots, m, \tag{38}$$

$$\delta_0^+ \leq \epsilon, \qquad \delta_0^- \leq \epsilon. \tag{39}$$

Note that the above problem is reduced to the Chebyshev primal master problem when $\epsilon = 0$. For a given ϵ, we solve the above problem by using the procedure ChebyshevCenterColGen (or PAChebyshevCenterColGen). We then reduce the value of ϵ and solve the above problem again by taking the last dual solution of the previous problem as the initial dual bound solution and the stabilization center of the new problem to be solved. The algorithm is terminated when $\epsilon = 0$.

4. Computational experiments

In this section, we present the computational results which demonstrate the effectiveness of the Chebyshev center based column generation algorithm. Since our aim is to accelerate the column generation procedure, no efforts were made to refine the branch-and-price procedure for obtaining the optimal integer solutions, i.e., we only solved the problem at the root node of the branch-and-price search tree. The experiments were conducted on three well-known problem classes, namely, the binpacking problem, the vehicle routing problem with time windows (VRPTW), and the generalized assignment problem (GAP). We solved the test problems with six different column generation algorithms as follows:

- Chebyshev center based column generation (CG) (`Chebyshev`) with $\alpha = 1$.
- Proximity adjusted Chebyshev center based CG (`PA Chebyshev`) with $T = 10$ and $\Omega = 100$.
- Stabilized column generation (`Stabilization`).
- Standard column generation (`Kelley`).
- Stabilized Chebyshev algorithm (`Chebyshev+Sta.`).
- Stabilized proximity adjusted Chebyshev algorithm (`PA Chebyshev+Sta.`) with $T = 10$ and $\Omega = 100$.

The performances of the algorithms are compared by the performance profile graphs proposed in [13]. For a set of algorithms S and a set of problems P, we define the *performance ratio*

$$r_{p,s} = \frac{t_{p,s}}{\min\limits_{s\in S}\{t_{p,s}\}}, \quad \text{for all } p \in P, \tag{40}$$

where $t_{p,s}$ is any performance measure (e.g., iteration number, time) of algorithm $s \in S$ for problem $p \in P$. We, then, define $\rho_s(\tau)$ as the probability for algorithm $s \in S$ that a performance ratio is within a factor τ of the best possible ratio:

$$\rho_s(\tau) = \frac{|\{p \in P \mid r_{p,s} \le \tau\}|}{|P|}, \tag{41}$$

where $|\cdot|$ represents the number of elements of a set. For any given $\tau \ge 1$, algorithms having large $\rho_s(\tau)$ are to be preferred; specifically $\rho_s(1)$ means the probability that the algorithm s will not be outperformed by the rest of the algorithms. A performance profile graph is obtained by plotting the probability $\rho_s(\tau)$ with varying τ. In the column generation procedure, the overall performance largely depends on the choice of the column generation subproblem, i.e., *oracle*. Here we report the performance profile graphs for the test problems by using the iteration number as the performance measure because the iteration number is closely related to the depth (goodness) of cuts, and independent of any specific algorithm (and the implementation) of the oracle.

In the algorithms using the stabilization technique, (`Stabilization`, `Chebyshev+Sta.`, and `PA Chebyshev+Sta.`), the penalty coefficient ϵ was initially set to 0.1, and then sequentially updated to 0.01, 0.001, 0.0001, and 0. The convergence tolerances for the Chebyshev algorithms, ε, and the reduced cost tolerances of the column generation subproblem were 10^{-4}. We used the L_1 norm in calculating the Chebyshev center. All computational tests presented here were performed on an AMD X2 2.9 GHz PC with 4 GB RAM. The implementations of the algorithms were done with C# using `CPLEX` 10.1 as a linear programming solver.

4.1. Binpacking problem

The binpacking problem is to minimize the number of bins of width L that are needed to pack all items $i = 1, \dots, I$ of widths $w_1, \dots, w_I$. The linear programming relaxation of the standard covering type Dantzig–Wolfe decomposition model can be given as

(Binpacking Primal)

$$\begin{aligned} \min \quad & \sum_{p\in P} x_p \\ \text{s.t.} \quad & \sum_{p\in P} a_{ip} x_p \ge 1, \quad \forall i = 1, \dots, I, \\ & x_p \ge 0, \quad \forall p \in P, \end{aligned}$$

(Binpacking Dual)

$$\begin{aligned} \max \quad & \sum_{i=1}^{I} \pi_i \\ \text{s.t.} \quad & \sum_{i=1}^{I} a_{ip} \pi_i \le 1, \quad \forall p \in P, \\ & \pi_i \ge 0, \quad \forall i = 1, \dots, I, \end{aligned}$$

where P is the set of possible packing patterns of a bin, i.e., a set of feasible solutions of $\sum_{i=1}^{I} w_i a_i \le L,\ a_i \in \{0, 1\},\ \forall i = 1, \dots, I$. The coefficient a_{ip} is 1 if item i is packed by pattern p, 0 otherwise. The decision variable x_p indicates the number of bins using packing pattern p.

From the above dual problem, we can easily derive a feasible dual solution of $\tilde{\pi}_i = w_i/L,\ i = 1, \dots, I$ by inspection. We used this solution as an initial stabilization center for the stabilized column generation algorithm and as an initial best dual solution for the Chebyshev center column generation algorithm. Let P' denote a subset of P. The Chebyshev primal master problem can be formulated as follows:

$$\text{(Binpacking Chebyshev Primal Master):} \quad \begin{aligned} \min \quad & \sum_{p\in P'} x_p - \sum_{i=1}^{I} \tilde{\pi}_i z \\ \text{s.t.} \quad & \sum_{p\in P'} a_{ip} x_p - y_i - z \ge 0, \quad \forall i = 1, \dots, I, \\ & \sum_{p\in P'} \|a_p\| x_p + \sum_{i=1}^{I} y_i + \alpha \|\vec{1}\| z \ge 1, \\ & x_p \ge 0, \quad \forall p \in P', \qquad y_i \ge 0, \quad \forall i = 1, \dots, I, \qquad z \ge 0, \end{aligned}$$

where $\vec{1}$ is an I dimensional vector of all ones and $a_p := (a_{ip})_{i=1}^{I}$.

Note that the column generation subproblem is unchanged from the classical column generation. Let $\pi_i,\ \forall i = 1, \dots, I$ denote the dual variables associated with the first inequalities of (Binpacking Chebyshev Primal Master). The column

Table 1
Binpacking problem.

Prob	Chebyshev		PA Chebyshev		Chebyshev+Sta.		PA Chebyshev+Sta.		Stabilization		Kelley	
	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)
u120	373.1	0.4 (35.9%)	360.5	0.4 (32.4%)	**201.8**	0.3 (34.4%)	244.4	0.3 (35.7%)	329.8	0.4 (33.3%)	403.2	0.3 (31.0%)
u250	760.6	2.2 (31.5%)	727.1	2.4 (27.6%)	**398.8**	1.8 (31.8%)	576.6	1.9 (28.6%)	669.9	2.2 (33.9%)	834.6	1.7 (30.2%)
u500	1437.6	21.1 (52.0%)	1388.3	20.0 (51.9%)	**797.1**	14.8 (66.6%)	1154.1	17.0 (52.8%)	1222.5	15.7 (42.5%)	1584.0	10.0 (18.0%)
u1000	2792	1274.7 (92.6%)	2721	1271.2 (92.8%)	**1614**	824.4 (96.9%)	2303	873.7 (91.8%)	2346	175.0 (59.1%)	3073	82.6 (9.6%)
t60	268.6	0.3 (66.7%)	250.3	0.3 (66.7%)	100.9	0.1 (44.4%)	**94.1**	0.1 (42.9%)	99.8	0.1 (50.0%)	213.3	0.1 (50.0%)
t120	483.8	3.2 (87.9%)	445.9	3.2 (89.1%)	**177.5**	0.4 (52.3%)	200.7	0.5 (68.8%)	225.2	0.7 (58.0%)	405.0	0.5 (51.9%)
t249	819.8	15.2 (86.9%)	741.7	14.9 (85.6%)	**370.6**	2.3 (48.9%)	494.3	3.5 (63.4%)	487.5	11.2 (84.8%)	809.8	3.2 (52.4%)
t501	1564.9	20.4 (30.1%)	1392.0	16.6 (31.0%)	**757.6**	14.6 (66.8%)	964.3	17.9 (61.7%)	1010.8	393.6 (98.0%)	1594.0	18.0 (39.6%)

generation subproblem corresponds to the 0–1 knapsack problem, which finds a profitable packing pattern:

$$\text{(Binpacking Oracle):}\quad \begin{aligned} \max\quad & \sum_{i=1}^{I} \pi_i a_i \\ \text{s.t.}\quad & \sum_{i=1}^{I} w_i a_i \le L, \\ & a_i \in \{0,1\}, \quad \forall i = 1,\dots,I. \end{aligned}$$

4.1.1. *Test instances*

The test problems were taken from the OR-library [3], which has two classes of instances: The first class, u-problems, consists of problems with $L = 150$ and the widths of items are uniformly distributed in [20, 100]. The second class, t-problems, are *triplets*, which means that triples of items are to be packed exactly into bins of $L = 100$. The problem set includes problems for different numbers of items: 120, 250, 500, and 1000 for the u-class problems, and 60, 120, 249, and 501 for the t-class problems. There are 20 problems for each problem class with a given number of items. Therefore the results were averaged over 20 problems for each problem type (except for some problems in u1000 which could not be solved in the 1 h time limit.) To solve the knapsack subproblem, we used Horowitz and Sahni's branch and bound algorithm [25].

4.1.2. *Comparisons with other algorithms*

Table 1 reports the computational results of the six different algorithms for the binpacking problems. The headings `#iter`, `time`, and `sub%` stand for the number of iterations, total cpu time (in seconds) spent in the column generation procedure, and the percentage of spent time in solving the (knapsack) subproblem, respectively. The bold numbers represent the best performance in terms of the iteration number. Fig. 4 shows the performance profile graph for the binpacking problems. The `PA Chebyshev` algorithm consistently performs better than the `Chebyshev` algorithm, while `Kelley's` algorithm outperforms the `Chebyshev` algorithm when $\tau > 2$, which implies that there are some problem instances the `Chebyshev` algorithm performs very poorly. The `Chebyshev+Sta.` algorithm can clearly be seen to have the largest probability that it will perform the best of all six algorithms for every range of τ. An interesting observation is that the `Chebyshev+Sta.` algorithm performs better than the `PA Chebyshev+Sta.` algorithm. This behavior is understandable if we note that adjusting the proximity parameter α may cancel the stabilization effect. In other words, we want to move the Chebyshev center by adjusting α, so we may lose the stabilization effect on the dual solution.

4.2. *VRP with time windows*

The VRPTW is to visit a given set of customers I exactly once with available vehicles while taking into account the capacities of the vehicles. The objective is to minimize the total traveling distances and each customer's time window should be observed when a vehicle visits the customer. Let R denote the set of all feasible routes that satisfy the customers' time windows and vehicle capacity. Any route of a vehicle should start from and end at the depot, and the problem allows the vehicle to wait at any customer location until service is possible. The linear programming relaxation of the standard Dantzig–Wolfe decomposition model can be stated as

$$\begin{array}{ll|ll} & \text{(VRPTW Primal)} & & \text{(VRPTW Dual)} \\ \min & \displaystyle\sum_{r\in R} c_r x_r & \max & \displaystyle\sum_{i\in I} \pi_i \\ \text{s.t.} & \displaystyle\sum_{r\in R} \delta_{ir} x_r \ge 1, \quad \forall i \in I, & \text{s.t.} & \displaystyle\sum_{i\in I} \delta_{ir}\pi_i \le c_r, \quad \forall r \in R, \\ & x_r \ge 0, \quad \forall r \in R, & & \pi_i \ge 0, \quad \forall i \in I, \end{array}$$

where c_r is the distance of route $r \in R$. The indicator δ_{ir} is the number of stops at customer i in route r. Note that our definition of a route permits cycles in the route as long as the time windows and capacity constraints are satisfied; therefore a route may visit the same customer multiple times. Provided that the triangle inequalities are satisfied, it can be shown that

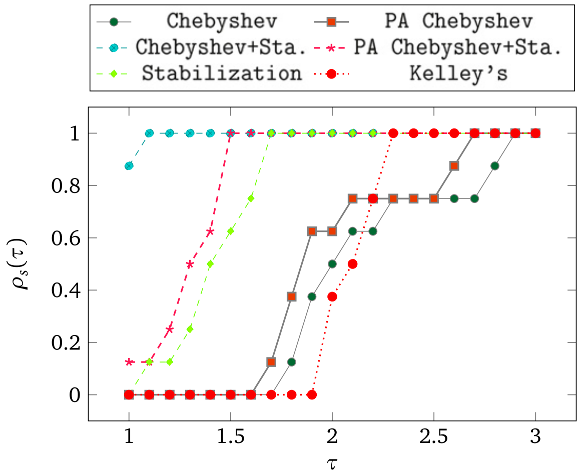


Fig. 4. Performance profile graph for the binpacking problems.

there exists an integer optimal solution without a cycle [10]. The decision variable x_r determines how many times route r should be operated, and it is easily seen that $x_r \leq 1$ at the optimality. From the above dual problem, a feasible dual solution of $\tilde{\pi}_i = \min_{(j,i)\in\theta_i} c_{ij}$ is easily derived by inspection, where θ_i is the set of edges incident to node i. We used $\tilde{\pi}_i$ as an initial stabilization center for the stabilized column generation algorithm, and an initial best dual solution for the Chebyshev center column generation algorithm. Let R' denote a subset of R. Then, the Chebyshev primal master problem can be formulated as follows:

$$\text{(VRPTW Chebyshev Primal Master):}\quad \begin{aligned} \min\quad & \sum_{r\in R'} c_r x_r - \sum_{i\in I} \tilde{\pi}_i z \\ \text{s.t.}\quad & \sum_{r\in R'} \delta_{ir} x_r - y_i - z \geq 0, \quad \forall i \in I, \\ & \sum_{r\in R'} \|\delta_r\| x_r + \sum_{i\in I} y_i + \alpha \|\vec{1}\| z \geq 1, \\ & x_r \geq 0, \quad \forall r \in R', \qquad y_i \geq 0, \quad \forall i \in I, \qquad z \geq 0, \end{aligned}$$

where $\vec{1}$ is a $|I|$ dimensional vector of all ones.

The column generation subproblem corresponds to the shortest path problem with resource constraints (SPPRC) [21], which finds a profitable route such that $\sum_{i\in I} \delta_{ir}\pi_i > c_r$. This problem can be solved by the labeling algorithm, as proposed by Irnich and Desaulniers [21] and Dror [14].

4.2.1. Test instances

The test problems shown here are taken from the well-known Solomon problems [32] and classified into three classes of problems: whose in which the customers are distributed randomly over the square (R class) or, alternatively, clustered (C class), and those of consisting of a combination of randomly placed and clustered customers (RC class). The number of customers is 100 in all cases. Problems which could not be solved within the 1-hour time limit are excluded.

4.2.2. Comparisons with other algorithms

Table 2 compares the performances of the column generation methods, and Fig. 5 shows the performance profile graph on the VRPs. Note that we used a logarithm scale for the horizontal axis because some of the performance ratios are greater than 10. Again, the `Chebyshev+Sta.` algorithm performs the best, and the `Stabilization` algorithm performs better than `Kelley's` algorithm for every type of problem. However, the performances of the other algorithms vary considerably depending on the type of problem. In particular, the `Chebyshev` algorithm seems to be more effective than the `PA Chebyshev` algorithm for the C class problems, while the `Chebyshev` algorithm and the `PA Chebyshev` algorithm show comparable performances for the R and RC class problems. `Kelley's` algorithm can be seen to perform better than the `Chebyshev` algorithm (or the `PA Chebyshev` algorithm) for the R and RC class problems. This behavior may be interpreted as follows. In the C class problems, the customers are clustered, so there may be many vehicle routes of similar distances. It is fairly well known that the standard column generation method simply collapses for highly degenerate problems, i.e., it may produce many similar columns without any improvement of the objective value. In other words, as there are many nearly equivalent extreme points in the dual polyhedron, any nonextreme point based method such as the Chebyshev algorithm may be more effective. Another possible explanation of this behavior is that the extreme dual optimal solutions for the customers, π_i, have many uneven values, thereby yielding routes serving only a portion of the clustered customers. As the preferred route is to serve clustered customers by the same vehicle, it may be advantageous if the dual solutions have even (nonzero) values for the clustered customers, which is achievable by centering or stabilizing the dual solution.

Table 2
Vehicle routing problem.

Prob	Chebyshev		PA Chebyshev		Chebyshev+Sta.		PA Chebyshev+Sta.		Stabilization		Kelley	
	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)
C101	509	12.3 (96.2%)	634	21.5 (96.0%)	**137**	2.7 (97.0%)	762	23.3 (96.5%)	273	8.5 (97.7%)	825	35.0 (97.9%)
C102	935	256.6 (99.7%)	1219	459.1 (99.3%)	**220**	62.6 (99.8%)	902	323.6 (99.3%)	526	201.6 (99.7%)	1084	447.8 (99.7%)
C105	685	25.8 (97.9%)	814	46.8 (97.1%)	**140**	4.8 (98.3%)	1174	73.0 (96.6%)	315	17.5 (98.5%)	1653	107.8 (97.8%)
C106	578	74.5 (99.4%)	814	126.2 (99.0%)	**176**	9.7 (98.9%)	856	148.3 (98.9%)	443	81.4 (99.6%)	1062	199.9 (99.5%)
C107	799	41.7 (98.4%)	1210	99.2 (97.1%)	**173**	6.7 (98.5%)	1147	97.2 (97.6%)	376	36.3 (99.1%)	1036	142.1 (99.2%)
C108	599	351.7 (99.9%)	703	584.4 (99.8%)	**325**	165.2 (99.9%)	695	497.3 (99.8%)	536	390.4 (99.9%)	669	678.4 (99.9%)
C109	563	1070.4 (100.0%)	661	2252.4 (99.9%)	**374**	553.5 (99.9%)	625	1641.8 (99.9%)	547	1578.1 (100.0%)	666	2113.5 (100.0%)
R101	428	4.3 (93.0%)	486	5.9 (88.3%)	**284**	2.9 (93.1%)	451	5.3 (88.9%)	401	4.7 (93.0%)	466	5.3 (94.0%)
R102	578	24.6 (98.0%)	553	30.3 (97.1%)	**369**	16.8 (98.4%)	509	26.5 (97.1%)	489	27.4 (98.3%)	573	29.8 (98.6%)
R103	1075	174.1 (99.2%)	594	97.1 (98.7%)	**454**	61.9 (99.2%)	657	107.3 (98.3%)	543	99.5 (99.3%)	617	115.4 (99.5%)
R105	553	19.2 (97.0%)	573	27.4 (94.1%)	**381**	13.8 (97.1%)	580	27.8 (94.2%)	492	21.3 (97.0%)	592	27.0 (97.7%)
R106	1215	138.0 (98.7%)	618	101.8 (98.2%)	**409**	57.5 (99.2%)	623	101.4 (98.2%)	566	104.0 (99.3%)	642	124.5 (99.4%)
R107	1060	361.2 (99.6%)	668	351.2 (99.2%)	**397**	152.3 (99.7%)	647	309.5 (99.2%)	568	337.2 (99.8%)	693	370.1 (99.8%)
R109	514	57.3 (99.1%)	536	97.7 (98.8%)	**360**	45.1 (99.1%)	546	101.1 (98.6%)	487	84.4 (99.3%)	580	110.1 (99.5%)
R110	562	188.4 (99.7%)	597	311.2 (99.5%)	**412**	149.3 (99.7%)	572	272.0 (99.4%)	546	291.3 (99.8%)	592	325.1 (99.8%)
R111	739	194.2 (99.6%)	632	242.2 (99.2%)	**383**	88.8 (99.5%)	653	242.0 (99.1%)	548	231.8 (99.7%)	660	274.4 (99.7%)
RC101	457	10.9 (96.6%)	503	14.3 (94.0%)	**365**	9.8 (96.6%)	499	14.0 (93.6%)	448	12.6 (96.4%)	495	13.6 (97.1%)
RC102	598	62.7 (99.1%)	564	84.6 (98.7%)	**394**	58.5 (99.4%)	557	91.5 (98.7%)	492	92.7 (99.4%)	571	102.7 (99.5%)
RC103	677	274.3 (99.7%)	618	343.6 (99.5%)	**410**	268.9 (99.8%)	583	319.6 (99.6%)	510	339.1 (99.8%)	580	354.1 (99.8%)
RC105	519	30.8 (98.6%)	519	38.5 (97.8%)	**354**	24.0 (98.6%)	529	40.3 (97.5%)	454	38.2 (98.8%)	547	41.0 (98.9%)
RC106	470	64.6 (99.3%)	520	97.5 (99.0%)	**377**	58.8 (99.3%)	523	98.3 (99.1%)	475	89.7 (99.5%)	538	98.0 (99.5%)
RC107	523	382.3 (99.9%)	537	485.6 (99.8%)	**373**	271.8 (99.9%)	535	558.8 (99.8%)	478	498.1 (99.9%)	553	532.1 (99.9%)

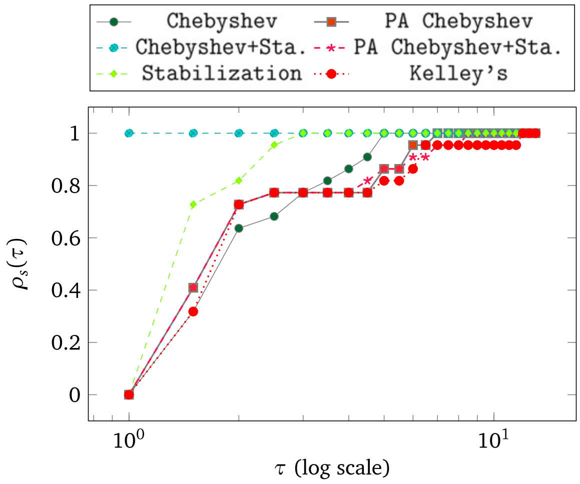


Fig. 5. Performance profile graph for the VRP problems.

4.3. Generalized assignment problem

The generalized assignment problem (GAP) is to obtain a minimum cost assignment plan of a set of jobs J to a set of agents I, while taking the capacities of the agents into account. Every job should be assigned to exactly one agent. Job $j \in J$ consumes capacity a_{ij} and incurs cost c_{ij} when assigned to agent $i \in I$, whose capacity is b_i. Let K_i be the set of all possible assignment patterns for agent i. The linear programming relaxation of the standard Dantzig–Wolfe decomposition model can be addressed as

$$\begin{aligned} &\text{(GAP Primal)} \\ \min\ & \sum_{i\in I}\sum_{k\in K} c_k^i x_k^i, \\ \text{s.t.}\ & \sum_{i\in I}\sum_{k\in K_i} \delta_k^j x_k^i \geq 1, \quad j \in J, \\ & -\sum_{k\in K_i} x_k^i \geq -1, \quad \forall i \in I, \\ & x_k^i \geq 0, \quad \forall k \in K_i, i \in I, \end{aligned} \qquad \begin{aligned} &\text{(GAP Dual)} \\ \max\ & \sum_{j\in J} \pi_j - \sum_{i\in I} \phi_i, \\ \text{s.t.}\ & \sum_{j\in J} \delta_k^j \pi_j - \phi_i \leq c_k^i, \quad \forall k \in K_i, i \in I, \\ & \pi_j \geq 0, \quad \forall j \in J, \\ & \phi_i \geq 0, \quad \forall i \in I, \end{aligned}$$

where δ_k^j is 1 if job j is in assignment pattern $k \in K_i$ for agent i, and c_k^i is defined as $\sum_{j\in J} c_{ij}\delta_k^j$ for $k \in K_i$ and $i \in I$. By inspection, we set the initial dual solution as $\tilde{\pi}_j = \min_{i\in I} c_{ij}$, $\forall j \in J$ and $\tilde{\phi}_i = 0$, $\forall i \in I$. Let K_i' denote a restricted set of K_i for agent i. Then, the Chebyshev primal master problem can be formulated as follows:

$$\text{(GAP Chebyshev Primal Master):} \quad \begin{aligned} \min\ & \sum_{i\in I}\sum_{k\in K_i'} c_k^i x_k^i - \left(\sum_{j\in J}\tilde{\pi}_j - \sum_{i\in I}\tilde{\phi}_i\right) z \\ \text{s.t.}\ & \sum_{i\in I}\sum_{k\in K_i'} \delta_k^j x_k^i - u_j - z \geq 0, \quad \forall j \in J, \\ & -\sum_{k\in K_i'} x_k^i - v_i + z \geq 0, \quad \forall i \in I, \\ & \sum_{i\in I}\sum_{k\in K_i'} \|(\delta_k, -1)\| x_k^i + \sum_{j\in J} u_j + \sum_{i\in I} v_i + \alpha\|\vec{1}\| z \geq 1, \\ & x_k^i \geq 0, \quad \forall i \in I, k \in K_i', \\ & u_j \geq 0, \quad \forall j \in I, \qquad v_i \geq 0, \quad \forall i \in I, \qquad z \geq 0, \end{aligned}$$

where $\vec{1}$ is a $|I| + |J|$ dimensional vector of all ones, and $\delta_k := (\delta_k^j)_{j\in J}$.

The column generation subproblem corresponds to the 0–1 knapsack problem, which finds a beneficial assignment pattern such that $\sum_{j\in J}(\pi_j - c_{ij})\delta_j > \phi_i$ for each agent $i \in I$:

$$\text{(GAP Oracle):} \ \max \sum_{j\in J}(\pi_j - c_{ij})\delta_j, \quad \text{s.t.} \sum_{j\in J} a_{ij}\delta_j \leq b_i,\ \delta_j \in \{0, 1\}, \quad \forall j \in J.$$

At any column generation iteration, we solve GAP oracles for all agents, and all assignment patterns with $\sum_{j\in J}(\pi_j - c_{ij})\delta_j > \phi_i$ are added to the master problem.

Table 3
Generalized assignment problem.

Prob	Chebyshev		PA Chebyshev		Chebyshev+Sta.		PA Chebyshev+Sta.		Stabilization		Kelley's	
	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)	#iter	time (sub%)
d05100	985	34.6 (57.9%)	861	38.4 (56.3%)	729	23.9 (54.7%)	791	34.7 (60.6%)	**712**	16.1 (58.9%)	825	26.4 (58.8%)
d10100	233	3.7 (50.9%)	**203**	3.6 (53.3%)	216	2.4 (39.1%)	229	4.2 (46.0%)	302	4.3 (49.1%)	267	3.8 (50.3%)
d10200	1104	350.6 (49.9%)	1022	518.9 (37.7%)	**683**	67.5 (33.5%)	1013	483.5 (23.3%)	916	130.5 (46.2%)	1114	225.1 (45.8%)
d20100	126	1.4 (44.6%)	**124**	1.5 (43.5%)	154	1.6 (39.9%)	136	2.0 (47.3%)	176	2.0 (57.4%)	162	1.8 (54.9%)
d20200	328	34.7 (39.6%)	**263**	35.5 (43.0%)	396	38.5 (24.8%)	282	35.6 (38.9%)	445	44.2 (33.1%)	411	42.9 (33.6%)
e05100	709	19.8 (55.3%)	675	26.8 (49.8%)	692	15.3 (46.5%)	681	22.8 (57.3%)	**617**	12.7 (59.0%)	710	21.2 (57.1%)
e10100	276	4.1 (49.9%)	**236**	5.2 (41.8%)	347	4.6 (44.5%)	252	4.9 (49.0%)	319	4.5 (48.5%)	271	3.5 (48.1%)
e10200	1338	469.7 (39.0%)	1173	734.0 (28.3%)	1260	407.4 (30.2%)	1149	671.8 (24.9%)	**950**	132.8 (55.0%)	1267	247.9 (47.3%)
e20100	151	1.6 (52.1%)	**126**	1.8 (52.0%)	227	2.7 (43.5%)	151	2.6 (51.9%)	189	2.2 (53.9%)	168	1.7 (56.1%)
e20200	377	68.1 (61.1%)	**292**	66.6 (63.3%)	507	65.7 (39.1%)	299	51.9 (48.4%)	495	45.8 (35.6%)	439	37.7 (32.5%)

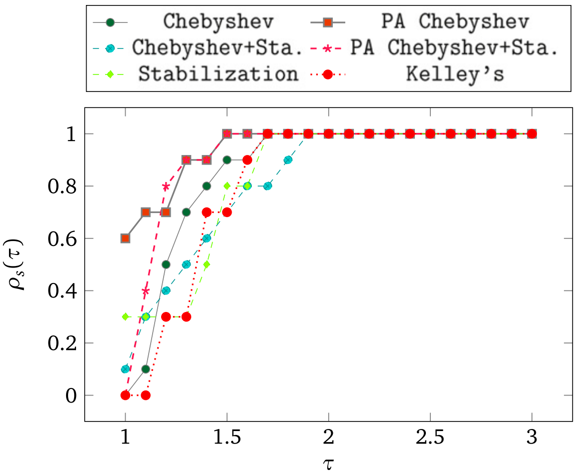


Fig. 6. Performance profile graph for the GAP problems.

4.3.1. *Test instances*

The test problems are taken from the OR-library [3], which contains five classes of problem instances. In our trials we used type D and E instances. Each problem's name represents the type of the problem, the number of agents, and the number of jobs. For example, d05100 stands for the problem of type D, which has five agents and 100 jobs to be assigned. To solve the knapsack subproblem, we used Horowitz and Sahni's branch and bound algorithm [25].

4.3.2. *Comparisons with other algorithms*

The performances of the column generation methods and the performance profile graph for the GAP problems are shown in Table 3 and Fig. 6, respectively. The `PA Chebyshev` algorithm outperforms the other five algorithms, while the performance gap is not apparent compared with the results of the binpacking and the VRP problems. The stabilized Chebyshev algorithms show inferior performances compared with the pure Chebyshev algorithms, which may be due to the relatively poorer performance of the `Stabilization` algorithm.

5. Conclusion

The column generation procedure based on the simplex algorithm often shows desperately slow convergence. In this paper, we propose the use of the Chebyshev center based column generation scheme to accelerate convergence. The Chebyshev center is the deepest point inside the polyhedron, and it can be obtained by reformulating the dual formulation of the master problem. The motivation for developing this approach is to prevent the so-called zig–zag behavior of dual solutions by being able to follow some center points of the dual polyhedron. To this end, we propose the proximity adjusted Chebyshev center algorithm in which the distance between the Chebyshev center and the dual bound inequality is dynamically adjusted, as this consistently showed better performance than the pure Chebyshev center algorithm. We also demonstrate here that the proposed algorithm can be used in a combination with the stabilization algorithm (stabilized Chebyshev algorithm).

We conducted computational experiments on the binpacking problem, the vehicle routing problem, and the generalized assignment problem. Our comparisons between six different column generation algorithms showed that the proposed algorithm can accelerate the column generation procedure.

One interesting observation is that on occasion the number of iterations and computational time do not appear to be correlated. When the times spent on solving the oracle problems during the column generation algorithm are more closely examined, they can be seen to fluctuate greatly at a number of specific iterations, indicating that the overall computational time may be highly dependent on the algorithm and the implementation of the oracle. It is, however, not easy to determine which conditions cause the oracle to perform poorly. Briant et al. [9] observed a similar behavior in their comparison of the classical column generation and the stabilized technique. The authors cautiously suggested that solving the oracles for the stabilized techniques may be more difficult due to the many nonzeros in the dual solutions. They also proposed that a warm-staring of the oracles might be advantageous because the dual solutions are supposedly close together in the stabilized techniques.

To use the proximity adjusted Chebyshev algorithm, one has to determine the values of two important parameters (T and Ω). It is barely imaginable that any single combination of parameter values performs best for any type of problem. It is also not easy to determine the best parameter values in advance since whose may depend on the specific properties of the problem to be solved. This difficulty in choosing the proper parameter values (defining of the penalty function) is also shared with the stabilization algorithms. We note that the choice of norm did not make a significant differences, but with the aim of maintaining numerical stability, the L_1 norm would appear to be a good choice.

Acknowledgments

The authors would like to thank several anonymous referees for their numerous and very helpful comments and suggestions.

References

[1] F. Babonneau, O. Du Merle, J.P. Vial, Solving large-scale linear multicommodity flow problems with an active set strategy and proximal-accpm, Operations Research 54 (1) (2006) 184–197.
[2] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance, Branch-and-price: column generation for solving huge integer programs, Operations Research 46 (3) (1998) 316–329.
[3] J. Beasley, Or-library: distributing test problems by electronic mail, Journal of Operations Research Society 41 (11) (1990) 1069–1072.
[4] W. Ben-Ameur, J. Neto, Acceleration of cutting-plane and column generation algorithms: applications to network design, Networks 49 (1) (2007) 3–17.
[5] H. Ben Amor, J. Desrosiers, A proximal trust-region algorithm for column generation stabilization, Computers & Operations Research 33 (4) (2006) 910–927.
[6] H. Ben Amor, J. Desrosiers, J.M.V. De Carvalho, Dual-optimal inequalities for stabilized column generation, Operations Research 54 (3) (2006) 454–463.
[7] B. Betrò, An accelerated central cutting plane algorithm for linear semi-infinite programming, Mathematical Programming 101 (3) (2004) 479–495.
[8] S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, UK, New York, 2004.
[9] O. Briant, C. Lemarechal, P. Meurdesoif, S. Michel, N. Perrot, F. Vanderbeck, Comparison of bundle and classical column generation, Mathematical Programming 113 (2) (2008) 299–344.
[10] A. Chabrier, Vehicle routing problem with elementary shortest path based column generation, Computers & Operations Research 33 (10) (2006) 2972–2990.
[11] G. Dantzig, P. Wolfe, The decomposition algorithm for linear programs, Econometrica 29 (4) (1961) 767–778.
[12] M. Desrochers, J. Desrosiers, M. Solomon, A new optimization algorithm for the vehicle-routing problem with time windows, Operations Research 40 (2) (1992) 342–354.
[13] E. Dolan, J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming 91 (2) (2002) 201–213.
[14] M. Dror, Note on the complexity of the shortest-path models for column generation in vrptw, Operations Research 42 (5) (1994) 977–978.
[15] O. Du Merle, J. Vial, Proximal accpm, a cutting plane method for column generation and lagrangian relaxation (application to the p-median problem), Technical report, HEX Geneve, Universite de Geneve, (2002.2.3).
[16] O. Du Merle, D. Villeneuve, J. Desrosiers, P. Hansen, Stabilized column generation, Discrete Mathematics 194 (1–3) (1999) 229–237.
[17] S. Elhedhli, J. Goffin, The integration of an interior-point cutting plane method within a branch-and-price algorithm, Mathematical Programming 100 (2) (2004) 267–294.
[18] J. Elzinga, T.G. Moore, A central cutting plane algorithm for the convex programming problem, Mathematical Programming 8 (1) (1975) 134–145.
[19] J. Goffin, J. Gondzio, R. Sarkissian, J. Vial, Solving nonlinear multicommodity flow problems by the analytic center cutting plane method, Mathematical Programming 76 (1) (1997) 131–154.
[20] J.L. Goffin, J.P. Vial, Convex nondifferentiable optimization: a survey focused on the analytic center cutting plane method, Optimization Methods & Software 17 (5) (2002) 805–867.
[21] S. Irnich, G. Desaulniers, Shortest Path Problems with Resource Constraints, Springer, 2004.
[22] J.E. Kelley, The cutting plane method for solving convex programs, Journal of the SIAM 8 (4) (1960) 703–712.
[23] M.E. Lubbecke, J. Desrosiers, Selected topics in column generation, Operations Research 53 (6) (2005) 1007–1023.
[24] R.E. Marsten, W.W. Hogan, J.W. Blankenship, The boxstep method for large-scale optimization, Operations Research 23 (3) (1975) 389–405.
[25] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, John Wiley & Sons, Inc., 1990.
[26] J. Mitchell, Cutting plane methods and subgradient methods, TutORials in Operations Research (2009) 34–61. (Chapter 2).
[27] A. Oukil, H. Ben Amor, J. Desrosiers, H. El Gueddari, Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems, Computers & Operations Research 34 (3) (2007) 817–834.
[28] A. Ouorou, A proximal cutting plane method using Chebychev center for nonsmooth convex optimization, Mathematical Programming (2008) 1–33.
[29] A. Pigatti, M. De Aragao, E. Uchoa, Stabilized branch-and-cut-and-price for the generalized assignment problem, Electronic Notes in Discrete Mathematics 19 (2004) 389–395.
[30] L.M. Rousseau, M. Gendreau, D. Feillet, Interior point stabilization for column generation, Operations Research Letters 35 (5) (2007) 660–668.
[31] M. Savelsbergh, A branch-and-price algorithm for the generalized assignment problem, Operations Research 45 (6) (1997) 831–841.
[32] M.M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Operations Research 35 (2) (1987) 254–265.
[33] F. Vanderbeck, Implementing mixed integer column generation, in: Column Generation, Kluwer Academic Publishers, Boston, MA, 2005.
[34] P. Wentges, Weighted Dantzig–Wolfe decomposition for linear mixed-integer programming, International Transactions in Operational Research 4 (2) (1997) 151–162.
[35] A. Westerlund, M. Gothe-Lundgren, T. Larsson, A stabilized column generation scheme for the traveling salesman subtour problem, Discrete Applied Mathematics 154 (15) (2006) 2212–2238.
[36] D. Yudin, A. Nemirovski, Informational complexity and efficient methods for the solution of convex extremal problems, Matekon 13 (1976) 3–25.