35/2008

Raport Badawczy

Research Report

RB/40/2008

An inexact bundle approach to cutting-stock problems

K.C. Kiwiel

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 3810100

fax: (+48) (22) 3810105

Kierownik Pracowni zgłaszający pracę:
Prof. dr hab. inż. Krzysztof C. Kiwiel

Warszawa 2008

An Inexact Bundle Approach to Cutting-Stock Problems

Krzysztof C. Kiwiel
Systems Research Institute, Newelska 6, 01–447 Warsaw, Poland, kiwiel@ibspan.waw.pl

We show that the LP relaxation of the cutting-stock problem can be solved efficiently by the recently proposed inexact bundle method. This method saves work by allowing inaccurate solutions to knapsack subproblems. With suitable rounding heuristics, our method solves almost all the cutting-stock instances from the literature.

Key words: nondifferentiable convex optimization, Lagrangian relaxation, integer programming, bundle methods, knapsack problems, cutting-stock
History: Initial version April 2005; revised May 2006; October 2006; September 2008.

---

1. Introduction

The classic Gilmore and Gomory (1961) formulation of the cutting-stock problem (CSP) is usually solved by LP-based column generation, rounding heuristics and branch-and-bound; see, e.g., (Belov and Scheithauer, 2002, 2006; Degraeve and Peeters, 2003; Degraeve and Schrage, 1999; Vance, 1998; Vanderbeck, 1999). Since column generation (CG) applied to its LP relaxation may converge slowly, there is interest in stabilized variants based on LP or QP (Ben Amor et al., 2004; Ben Amor and Valério de Carvalho, 2005; Briant et al., 2007). Alternatively, the highly efficient hybrid approach of Degraeve and Peeters (2003) generates additional columns by applying subgradient optimization to its Lagrangian relaxation.

In this paper we show that its LP relaxation can also be solved efficiently by the inexact bundle method of Kiwiel (2006a). This QP-based method saves work by allowing inaccurate solutions to Lagrangian subproblems. For the CSP, each subproblem is a knapsack problem (KP). We give a simple test for *inexact KP solutions* (see §2.2 below) that works well in practice for a standard branch-and-bound KP solver of Martello and Toth (1990). Further, to avoid the difficulties arising when a bounded KP is transformed into a 0–1 KP (Vanderbeck, 2002), we use *relaxed bounds*. Next, by adapting the ideas of (Belov and Scheithauer, 2002; Holthaus, 2002; Stadtler, 1990; Wäscher and Gau, 1996) to our inexact framework, we give rounding heuristics that solve *almost all* the CSP instances from the literature; in particular, they perform better than the best heuristics of Wäscher and Gau (1996). In effect, our inexact KP solutions, bound relaxation and rounding heuristics should be of interest also for other, more traditional CG-based approaches to the CSP.

We now provide a historical perspective for our contributions. Our work was inspired by Briant et al. (2007), where (together with four other applications) the LP relaxation of the CSP was solved by several variants of CG and a standard bundle method. On some CSP instances, bundle was much slower than CG, mostly because its subproblems were more difficult for the KP solver of Vanderbeck (2002). Hence Claude Lemaréchal suggested the CSP as a testing example for our inexact bundle (Kiwiel, 2006a). For technical reasons, instead of the KP solver of Vanderbeck (2002), we used the MT1R procedure of Martello and Toth (1990). Our initial quite disappointing results improved greatly once we used relaxed KP bounds and inexact solutions: our method became much faster in practice than *all* the algorithms tested in (Briant et al., 2007, §2.2) (see §5.8). Next, we collected more test instances and adapted some rounding heuristics from the literature. The main aim was to appraise our inexact bundle solutions: they are deemed accurate enough if the heuristics solve almost all instances.

We now summarize our findings on admissible inexactness. The relative accuracy in dual function evaluations is controlled by the tolerance ϵ_r of our KP solver (cf. §2.2). First, for $\epsilon_r = 0$ (i.e., exact bundle), the average computing times are much greater than those for $\epsilon_r = 10^{-5}$ (usually by factors of 30 or more), although the iteration numbers and the heuristic performance are almost the same. Second, the iteration numbers and timings are close for $\epsilon_r = 10^{-3}$, 10^{-4} and 10^{-5}; however, relative to $\epsilon_r = 10^{-5}$, our heuristics perform much worse for $\epsilon_r = 10^{-3}$, and just marginally worse for $\epsilon_r = 10^{-4}$. Third, further experiments (not reported here for brevity) gave very close results for $\epsilon_r = 10^{-5}$, 10^{-6}, 10^{-7} and 10^{-8}. To sum up, $\epsilon_r = 10^{-5}$ seems to be a good borderline choice. On the other hand, since in the CSP the gap between the primal value and the relaxed dual value is usually less than 1, and either rounding heuristics or branch-and-bound should "close" this gap, it may seem more appropriate to ensure a given *absolute* accuracy $\epsilon_a < 1$ in dual function evaluations (see §5.6.3). Quite suprisingly, our results for a fairly large $\epsilon_a = 0.01$ are very close to those for $\epsilon_r = 10^{-5}$, whereas for $\epsilon_a = 0.05$ our heuristics perform slightly worse.

We thus present the first successful application of our inexact bundle method. Our approach is also useful for the conic bundle variant of Kiwiel and Lemaréchal (2007).

The paper is organized as follows. In §2 we recall the classic CSP model of Gilmore and Gomory (1961) and introduce inexact KP solutions for its Lagrangian relaxation. Our rounding heuristics are given in §3 in a general form suitable for other CSP solvers. The inexact bundle method is reviewed in §4. Our computational results are presented in §5.

2. Lagrangian relaxation of the CSP

The one-dimensional cutting-stock problem (CSP) is to minimize the number of stock pieces of width W used to meet the demands d_i for items to be cut at their widths $w_i \in (0, W]$, for $i = 1, \ldots, m$. The bin-packing problem (BPP) is a special case of the CSP with unit demands.

2.1. The Gilmore-Gomory model

This classic model is formulated as follows. Denote the set of *cutting patterns* by

$$P := \{\, p \in \mathbb{Z}^m_+ : wp \leq W \,\}. \tag{1}$$

Let z_p be the number of times pattern p is used. The original model has the form

$$\min \sum_{p \in P} z_p \quad \text{s.t.} \quad \sum_{p \in P} p z_p \geq d, \;\; z \in \mathbb{Z}^{|P|}_+. \tag{2a}$$

For Lagrangian relaxation we augment this model with the redundant constraint

$$\sum_{p \in P} z_p \leq N, \tag{2b}$$

where N is an upper bound on the optimal value of (2a) (e.g., $N = \sum_i d_i$); this ensures boundedness of the *ground set* $Z := \{z \in \mathbb{Z}^{|P|}_+ : \sum_p z_p \leq N\}$. Relaxing the demand constraint $\sum_p p z_p \geq d$ with a price vector u yields the *Lagrangian* $L(z; u) := \sum_p z_p + u(d - \sum_p p z_p)$ and the *dual function*

$$\theta(u) := \min_{z \in Z} \left\{ L(z; u) = ud + \textstyle\sum_{p \in P} (1 - up) z_p \right\}. \tag{3}$$

The *Lagrangian subproblem* above may be solved by finding a solution $p(u)$ of the KP

$$p(u) \in \operatorname{Arg\,max}\{up : p \in P\} = \operatorname{Arg\,max}\{up : wp \leq W, p \in \mathbb{Z}^m_+\} \tag{4}$$

and taking $z_{p(u)} = N$ and $z_p = 0$ for $p \neq p(u)$ if $up(u) > 1$, $z = 0$ otherwise, thus producing

$$\theta(u) = ud + N\,[\,1 - up(u)\,]_- \,, \tag{5}$$

where $[\cdot]_- := \min\{\cdot, 0\}$. Let v_* and v_{LP} denote the optimal values of (2) and its LP relaxation, respectively. It is well known that v_{LP} coincides with the dual optimal value

$$\theta_* := \max\left\{ \theta(u) : u \in \mathbb{R}^m_+ \right\}. \tag{6}$$

Experiments show that $\check{u} := w/W$ is a good initial estimate of solutions to the Lagrangian dual (6) (Ben Amor and Valério de Carvalho, 2005, §4), (Briant et al., 2007, §2). In fact $\check{u}$ minimizes the relaxed dual function

$$\theta_{LP}(u) := ud + N[\, 1 - u\underline{p}(u)\,]_-, \tag{7}$$

where $\underline{p}(u)$ solves the LP relaxation of (4). (Since $\theta_{LP}(\check{u}) = \check{u}d \le v_* \le N$, we see that $-d = -N(\check{u}d/N)(d/\check{u}d)$ is a subgradient of the second term of (7) at $\check{u}$: $0 \in \partial\theta_{LP}(\check{u})$.)

2.2. Inexact KP solutions

To strengthen our relaxation, we may consider only *proper* patterns p such that

$$p \le b \quad \text{with} \quad b_i := \min\{\, d_i, \lfloor W/w_i \rfloor \,\}, \quad i = 1{:}\,m. \tag{8}$$

Indeed, adding the bound $p \le b$ to (1) and (4) does not change v_*, but it may raise v_{LP} (Nitsche et al., 1999). Then the CG subproblem (4) becomes a bounded KP, which can be turned into a 0–1 KP via the transformation of (Martello and Toth, 1990, §3.2). However, this transformation may duplicate solution representations, thus creating difficulties for 0–1 KP solvers (Vanderbeck, 2002). To avoid duplicates, we may use the *relaxed* bound

$$p \le b' \quad \text{with} \quad b'_i := 2^{\lceil \log_2(b_i+1) \rceil} - 1, \quad i = 1{:}\,m, \tag{9}$$

which corresponds to replacing d_i in (8) by the smallest number of the form $2^j - 1$ with $j \ge 1$ such that $2^j - 1 \ge d_i$ ($2d_i - 1$ in the worst case); the number of transformed variables is the same. We solve the transformed KP by a double precision version of the branch-and-bound procedure MT1R of Martello and Toth (1990). To reduce its work, we allow MT1R to find an approximate solution for a given *relative accuracy tolerance* ϵ_r. Namely, the backtracking step exits if $\zeta \ge (1-\epsilon_r)\bar{\zeta}$, where $\zeta := up$ for the incumbent p and $\bar{\zeta}$ is MT1R's upper bound on the optimal value $up(u)$. Hence, by (5), we have the accuracy estimates

$$\underline{\theta}(u) := ud + N(\,1 - \bar{\zeta}\,)_- \le \theta(u) \le \bar{\theta}(u) := ud + N\,(\,1 - \zeta\,)_-\,, \tag{10a}$$

$$\bar{\theta}(u) - \underline{\theta}(u) \le N(\,\bar{\zeta} - \zeta\,) \le N\epsilon_r\bar{\zeta}. \tag{10b}$$

For a normal exit with an optimal $p = p(u)$, we may replace $\bar{\zeta}$ by ζ and ϵ_r by 0 in (10).

As for our choice of MT1R, we add that Valério de Carvalho (2005) used MT1R as well, Belov and Scheithauer (2006) employed a similar branch-and-bound solver, whereas Vanderbeck (1999) and Briant et al. (2007) used the more specialized branch-and-bound solver of Vanderbeck (2002). On the other hand, Degraeve and Peeters (2003) employed a similar branch-and-bound solver but with prices multiplied by 10,000 and rounded to integers, without discussing the effects of inexact KP solutions. Further, more recent KP solvers (Kellerer et al., 2004) accept integer data only; hence their use with suitable price roundings is left open for a future study. To sum up, MT1R is outdated, but we could not find anything better, and we believe that the current results will serve as a useful yardstick for future work with modern KP solvers.

3. Heuristic rounding of relaxed solutions

Typical rounding heuristics for the CSP proceed as follows; cf. (Belov and Scheithauer, 2002, 2006; Degraeve and Peeters, 2003; Holthaus, 2002; Scheithauer et al., 2001; Stadtler, 1990; Wäscher and Gau, 1996). A solution $\hat{z}$ of the LP relaxation is rounded down into an integer solution $\bar{z} := \lfloor \hat{z} \rfloor$. Next, a sequential heuristic applied to the *residual* problem (2) with d replaced by $d' := d - \sum_p p\bar{z}_p$ delivers a residual solution $\tilde{z}$. Then the sum $\bar{z} + \tilde{z}$ serves as a possibly inexact solution of (2) (which is exact if its value is equal to a lower bound on v_*; e.g., $\lceil v_{LP} \rceil$). Since for simple rounding down ($\bar{z} = \lfloor \hat{z} \rfloor$), the residual problem may be too large to be solved optimally by a heuristic, some components of $\bar{z}$ may be increased (Holthaus, 2002; Scheithauer et al., 2001); however, if the residual problem becomes too small to produce a solution to the original problem, some components of $\bar{z}$ may be decreased (Belov and Scheithauer, 2002).

In §3.1 we give a general rounding procedure, which augments the ideas of Belov and Scheithauer (2002) and Holthaus (2002) with the oversupply reduction of Stadtler (1990). As for sequential heuristics, in §3.2 we describe minor (but useful) modifications of the first-fit-decreasing (FFD) of Chvátal (1983) and the heuristics of Belov and Scheithauer (2007) and Holthaus (2002). Since it pays to call lighter heuristics first, useful combinations of rounding and sequential heuristics are detailed in §3.3.

We add that the rounding procedures of (Vanderbeck, 1999, §3.7) and (Wäscher and Gau, 1996, RSUC) would be difficult to implement in our context. As for sequential heuristics, we also tried the best-fit-decreasing of Chvátal (1983) and the fill bin heuristics of Vanderbeck (1999), but they did not perform significantly better than FFD in our trials.

3.1. A general rounding procedure

Numbering the patterns so that $P = \{p^j\}_{j=1}^n$, we may write (2a) as

$$\min \sum_{j=1}^{n} z_j \quad \text{s.t.} \quad \sum_{j=1}^{n} p^j z_j \geq d, \;\; z \in \mathbb{Z}_+^n. \tag{11}$$

Given an incumbent solution z^* of (11) (e.g., found by FFD) and a point $\hat{z} \in \mathbb{R}_+^n$ (e.g., found by LP relaxation), the following procedure attempts to improve z^* by calling a heuristic on residual problems derived from rounded variants of $\hat{z}$. Let $e := (1, \dots, 1) \in \mathbb{R}^n$.

Procedure 1 (Rounding procedure).

Step 1 (*Rounding down*). Set $\bar{z} := \lfloor \hat{z} \rfloor$ and $d' := d - \sum_j p^j \bar{z}_j$. Sort the fractional parts $r_j := \hat{z}_j - \bar{z}_j$ so that $r_{j_1} \geq \dots \geq r_{j_n}$, and set $\bar{n} := |\{j : r_j > 0\}|$.

Step 2 (*Oversupply reduction*). While $d' \not\geq 0$, pick $\bar{\jmath}$ to maximize

$$\sum_{i:d'_i<0} w_i \min\{\, p_i^j, -d'_i \,\} \tag{12}$$

over j s.t. $\bar{z}_j > 0$, set $\bar{z}_{\bar{\jmath}} := \bar{z}_{\bar{\jmath}} - 1$ and $d' := d' + p^{\bar{\jmath}}$.

Step 3 (*Partial rounding up*). Set $I := \emptyset$. For $i = 1{:}\bar{n}$, if $p^{j_i} \leq d'$, set $\bar{z}_{j_i} = \bar{z}_{j_i} + 1$, $d' := d' - p^{j_i}$, $I := I \cup \{j_i\}$.

Step 4 (*Heuristic improvement*). Using a heuristic, find a feasible point $\tilde{z}$ for the residual problem (11) with d replaced by d'. If $e\bar{z} + e\tilde{z} < ez^*$, set $z^* := \bar{z} + \tilde{z}$.

Step 5 (*Residual problem extension*). If $I \neq \emptyset$, remove from I its last entry j, set $\bar{z}_j := \bar{z}_j - 1$, $d' := d' + p^j$ and return to Step 4.

If $\hat{z}$ solves the LP relaxation of an equality-constrained CSP, our procedure reduces to the one in (Belov and Scheithauer, 2002, §2.5); otherwise Step 2 (due to (Stadtler, 1990, Fig. 3)) helps. Following (Belov and Scheithauer, 2002, §5.2), our implementation allows at most ten returns from Step 5.

One of our heuristics uses the following modification of Step 3, based on the ideas in (Holthaus, 2002, §3.2).

Step 3' (*Partial rounding up*). Set $I := \emptyset$, $K := \{j : p^j \leq d', r_j > 0\}$. While $K \neq \emptyset$, pick $\bar{\jmath}$ to maximize $\sum_i p_i^j$ over $j \in K$, set $\bar{z}_{\bar{\jmath}} = \bar{z}_{\bar{\jmath}} + 1$, $d' := d' - p^{\bar{\jmath}}$, $I := I \cup \{\bar{\jmath}\}$, $K := \{j \in K : p^j \leq d', j \neq \bar{\jmath}\}$.

3.2. Sequential heuristics

We now describe our heuristics for the residual problem (2a) with d replaced by $d' \geq 0$. We assume that $w_1 \geq \ldots \geq w_m$.

Our implementation of FFD works as follows. Set $\tilde{z} := 0$, $d'' := d'$. While $d'' \neq 0$, generate the next pattern p by setting

$$p_i := \min\left\{ d''_i, \left\lfloor \left(W - \sum_{j<i} w_j p_j \right) / w_i \right\rfloor \right\} \quad \text{for } i = 1\colon m, \tag{13}$$

set $\kappa := \min\{\lfloor d''_i/p_i \rfloor : p_i > 0\}$, $\tilde{z}_p := \tilde{z}_p + \kappa$, $d'' := d'' - \kappa p$. The version of (Chvátal, 1983, p. 208) employs $\kappa \equiv 1$, and hence is less efficient for large demands.

Our modification of the *sequential heuristic procedure* (SHP) of (Holthaus, 2002, §3.2), given a price vector $\hat{u} \in \mathbb{R}^m$ (e.g., an approximate solution of (6)) and a *price tolerance* $u_{\text{tol}} > 0$ for rounding errors (we use $u_{\text{tol}} = 10^{-12}$), sets $\bar{u}_i := \max\{\hat{u}_i, u_{\text{tol}}\}$ for $i = 1\colon m$ and replaces the FFD formula (13) by the bounded KP

$$p \in \operatorname{Arg\,max}\{\, \bar{u}p : wp \leq W, p \leq d'', p \in \mathbb{Z}^m_+ \,\}. \tag{14}$$

Our implementation of the *sequential value correction* (SVC) heuristic of (Belov and Scheithauer, 2007, §2) records the best solution found by calling SHP at most thirty times with $\bar{u}$ modified as follows. Initially $\bar{u}_i := \max\{1, W\hat{u}_i\}$, $i = 1\colon m$. If $wd'' \not\leq W$, then after solving (14) and updating d'', for i such that $p_i > 0$, set

$$\bar{u}_i := [\, \gamma_i \bar{u}_i + (W/wp) w_i^{1.04} \,] / (\gamma_i + 1) \quad \text{with} \quad \gamma_i := \Omega_i \left(d'_i + d''_i \right) / p_i, \tag{15}$$

for Ω_i picked randomly in $[1/\Omega'_i, \Omega'_i]$, where Ω'_i is chosen at random in $[1, 1.5]$. An early exit occurs if SHP finds $\tilde{z}$ such that $e\bar{z} + e\tilde{z} = \lceil \theta(\hat{u}) \rceil$, in which case $z^* := \bar{z} + \tilde{z}$ is optimal.

3.3. Combinations of rounding and sequential heuristics

We now give more details on the five heuristics used in our experiments. The heuristics are described as if being called by a general solver for the LP relaxation of (11), which could be any variant of the CG procedure or the bundle method given in §4.

Our *initial* heuristic H0 calls FFD with $d' = d$ (i.e., on the original problem) to initialize the incumbent $z^* := \tilde{z}$, the upper bound $N := ez^*$ and the lower bound $\underline{\theta}_1 := -\infty$.

Suppose at iteration $k \geq 1$ of the solver, the following quantities are available: z^* is an incumbent solution of (11), $\hat{z}^k \in \mathbb{R}^n_+$ and $\hat{u}^k \in \mathbb{R}^m_+$ are tentative primal and dual solutions of the LP relaxation, and $\underline{\theta}_k$ is a lower bound on $\theta_* = v_{LP}$ (cf. (6)). If $ez^* = \lceil \underline{\theta}_k \rceil$, the solver may stop (since z^* is optimal). Otherwise, for iterations k specified below, the remaining heuristics consist in calling an extension of Procedure 1 with a copy of Step 4 inserted after Step 1; the sequential heuristics employed at these steps are listed below.

Our *periodic* heuristic H1 is called by the solver every twentieth iteration, starting from iteration $k = m+1$ (i.e., for $k = m+1, m+21, \ldots$), with the current relaxed solution $\hat{z} := \hat{z}^k$ and the lower bound $\underline{\theta}_k \le \theta_*$. H1 employs FFD in Procedure 1, exiting if $ez^* = \lceil \underline{\theta}_k \rceil$.

Our *final* heuristics H2, H3 and H4 are called successively upon termination of the solver, using the final $\hat{z} := \hat{z}^k$, $\hat{u} := \hat{u}^k$ and $\underline{\theta}_k$. H2 employs both FFD and SHP, H3 just SHP and the modified Step 3', whereas H4 uses SVC. Of course, H3 and H4 (or just H4) are not called if H2 (or H3) exits with $ez^* = \lceil \underline{\theta}_k \rceil$, whereas SVC exits when $e\bar{z} + e\tilde{z} = \lceil \underline{\theta}_k \rceil$. The impact of the various heuristics will be discussed in §5.7.

4. The inexact proximal bundle method

We now sketch the main features of the inexact bundle method of Kiwiel (2006a).

Our method generates *trial points* $u^k \in \mathbb{R}^m_+$, $k = 1, 2, \ldots$, at which the dual function θ is evaluated (possibly inexactly) as described in §2.2. Specifically, for each k, set p^k to the (possibly inaccurate) KP solution p satisfying the bounds of (10) for $u = u^k$, and let $\zeta_k := \zeta$, $\bar{\zeta}_k := \bar{\zeta}$. Recalling (3), define the associated *Lagrangian solution* z^k by setting

$$z^k_q := 0 \text{ for } q \neq p^k, \quad z^k_{p^k} := \begin{cases} N & \text{if } \zeta_k > 1, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Thus we have the lower bound $\underline{\theta}(u^k) \le \theta(u^k)$ and $L(z^k; u^k) = \bar{\theta}(u)$ in (10); in particular,

$$L(z^k; u^k) - \theta(u^k) \le N(\bar{\zeta}_k - \zeta_k) \le N\epsilon_r \bar{\zeta}_k. \tag{17}$$

Further, by (3), the following *linearization* of θ at u^k majorizes $\theta(u)$ for *all* u:

$$\theta_k(u) := L(z^k; u) = ud + \begin{cases} N(1 - up^k) & \text{if } z^k \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

Iteration k uses the polyhedral *cutting-plane model* of θ

$$\hat{\theta}_k(\cdot) := \min_{j \in J^k} \theta_j(\cdot) \quad \text{with} \quad k \in J^k \subset \{1, \ldots, k\} \tag{19}$$

for finding

$$u^{k+1} := \arg\max \left\{ \hat{\theta}_k(u) - \tfrac{1}{2t_k} |u - \hat{u}^k|^2 : u \in \mathbb{R}^m_+ \right\}, \tag{20}$$

where $t_k > 0$ is a *stepsize* that controls the size of $|u^{k+1} - \hat{u}^k|$ and the *prox center* $\hat{u}^k := u^{k'}$ has the value $\theta_{\hat{u}}^k := \theta_{k'}(u^{k'})$ for some $k' \leq k$ (usually $\theta_{\hat{u}}^k = \max_{j=1}^k \theta_j(u^j)$). Due to evaluation errors, we may have $\theta_{\hat{u}}^k > \hat{\theta}_k(\hat{u}^k)$, in which case the *predicted increase*

$$v_k := \hat{\theta}_k(u^{k+1}) - \theta_{\hat{u}}^k \tag{21}$$

may be nonpositive; then t_k is increased and u^{k+1} is recomputed to increase $\hat{\theta}_k(u^{k+1})$ until $v_k \geq |u^{k+1} - \hat{u}^k|^2/2t_k$. An *ascent* step to $\hat{u}^{k+1} := u^{k+1}$ with $k' := k+1$ occurs if

$$\theta_{k+1}(u^{k+1}) - \theta_{\hat{u}}^k \geq \kappa v_k \tag{22}$$

for a fixed $\kappa \in (0,1)$ (we use $\kappa = 0.1$). Otherwise, a *null* step $\hat{u}^{k+1} := \hat{u}^k$ improves the next model $\hat{\theta}_{k+1}$ with the new linearization θ_{k+1} as stipulated in (19).

If we omitted the quadratic term in (20), the resulting cutting-plane method could generate u^{k+1} far from the previous points, and it would require storing all linearizations ($J^k = \{1, \ldots, k\}$ in (19)). In contrast, the quadratic term usually keeps u^{k+1} close enough to the best point found so far, and it allows limiting the number of stored linearizations.

We solve subproblem (20) with the QP routine of Kiwiel (1994), which finds its multipliers $\{\nu_j^k\}_{j \in J^k} \subset \mathbb{R}_+$, also known as *convex weights*, such that $\sum_{j \in J^k} \nu_j^k = 1$ and the set $\hat{J}^k := \{j \in J^k : \nu_j^k \neq 0\}$ has at most $m+1$ elements. We set $J^{k+1} := J^k \cup \{k+1\}$ and then, if necessary, drop from J^{k+1} an index $j \in J^k \setminus \hat{J}^k$ with the largest $\theta_j(\hat{u}^k)$ to keep $|J^{k+1}| \leq M$ for a fixed $M \geq m+2$.

Combining the accumulated Lagrangian solutions $\{z^j\}_{j \in J^k}$ with their weights $\{\nu_j^k\}_{j \in J^k}$, we may estimate solutions to the LP relaxation of (2) via the *aggregate primal solution*

$$\hat{z}^k := \sum_{j \in J^k} \nu_j^k z^j. \tag{23}$$

In other words (cf. (16)), $\hat{z}_{p^j}^k = N\nu_j^k$ for nontrivial patterns p^j indexed by $J_P^k := \{j \in \hat{J}^k : z^j \neq 0\}$ (which need not be stored, since they can be recovered from $\nabla\theta_j = d - Np^j$; see (18)). Our heuristics also use the *lower bound* $\underline{\theta}_k := \max_{j \leq k} \underline{\theta}(u^j)$ on $\theta_* = v_{LP}$ (cf. (6)).

We now point out some useful consequences of the convergence analysis in (Kiwiel, 2006a, §5). The LP relaxation of (2) may be written as

$$v_{LP} := \min\ \bar{\psi}_0(z) := \sum_{p \in P} z_p \quad \text{s.t.} \quad \bar{\psi}(z) := d - \sum_{p \in P} p z_p \leq 0,\ z \in \operatorname{conv} Z. \tag{24}$$

Let $\epsilon := \sup_k[\theta_k(u^k) - \theta(u^k)]$ be the maximum evaluation error; by (17), we have $\epsilon \le \bar{\epsilon} := N\epsilon_r \sup_k \bar{\zeta}_k$. Consider the set of *$\epsilon$-optimal solutions* of the LP relaxation (24):

$$Z_\epsilon := \{\, z \in \operatorname{conv} Z : \bar{\psi}_0(z) \le v_{LP} + \epsilon, \bar{\psi}(z) \le 0 \,\}. \tag{25}$$

The limits $\theta_{\hat{u}}^\infty := \lim_k \theta_{\hat{u}}^k$, $\underline{\theta}_\infty := \lim_k \underline{\theta}_k$ satisfy $\theta_{\hat{u}}^\infty \in [v_{LP}, v_{LP} + \epsilon]$, $\underline{\theta}_\infty \in [\theta_{\hat{u}}^\infty - \bar{\epsilon}, v_{LP}]$, and there exists $K \subset \{1, 2, \ldots\}$ such that $\lim_{k \in K} \bar{\psi}_0(\hat{z}^k) = \theta_{\hat{u}}^\infty$ and $\overline{\lim}_{k \in K} \max_{i=1}^m \bar{\psi}_i(\hat{z}^k) \le 0$; in particular, the bounded sequence $\{\hat{z}^k\}_{k \in K}$ converges to the ϵ-optimal set Z_ϵ. If ϵ_r is small enough, the accuracy observed in practice corresponds to such estimates with ϵ and $\bar{\epsilon}$ determined by the maximum errors $\theta_k(u^k) - \theta(u^k)$ and $\theta(u^k) - \underline{\theta}(u^k)$ that occur for large k; since both errors are at most $N(\bar{\zeta}_k - \zeta_k)$, where the KP gap $\bar{\zeta}_k - \zeta_k$ is usually tiny for large k, small values of ϵ and $\bar{\epsilon}$ can be attained if the algorithm runs long enough.

We stop if $\min\{v_k, |\pi^k| + \alpha_k\} \le \epsilon_{\text{opt}}(1 + |\theta_{\hat{u}}^k|)$, where v_k is given by (21), $\pi^k := (\hat{u}^k - u^{k+1})/t_k$, $\alpha_k := v_k - t_k|\pi^k|^2$ and $\epsilon_{\text{opt}} > 0$ is an *optimality tolerance* (cf. (Kiwiel, 2006a, §4.2)). For $\epsilon_{\text{opt}} = \epsilon_r = 10^{-8}$, $\underline{\theta}_k$ usually agrees with θ_* in at least 8 digits, enough for our purposes.

5. Computational results

5.1. Data sets

In our computational experiments, for the CSP we use the 28 industrial instances of Vance (1998), the 10 industrial instances of Vanderbeck (1999), and the 20 industrial instances of Degraeve and Schrage (1999). In addition, we use the following randomly generated instances: the 4000 instances of Wäscher and Gau (1996), the 3360 instances of Degraeve and Peeters (2003) and the 120 instances of Vanderbeck (1999). For the BPP, we use the 540 randomly generated instances of Degraeve and Peeters (2003), and the 160 instances from the BINPACK collection of the OR-Library (Beasley, 1990).

The instances of Wäscher and Gau (1996) are constructed by the CUTGEN1 generator of Gau and Wäscher (1995), using the following parameter values: the number of orders $m = 10, 20, 30, 40, 50$, the width $W = 10{,}000$, the interval fraction $c = 0.25, 0.5, 0.75, 1$, and the average demand $\bar{d} = 10, 50$. The widths w_i are uniformly distributed integers between 1 and cW. For m uniform random numbers $R_1, \ldots, R_m \in (0, 1)$, the demands $d_i := \lfloor \frac{R_i m \bar{d}}{R_1 + \cdots + R_m} \rfloor$ for $i < m$, and $d_m := m\bar{d} - \sum_{i<m} d_i$ (in fact slightly more complicated formulas are used by Gau and Wäscher (1995)). Duplicate widths are aggregated by summing their demands.

Combining the different values for m, c and $\bar{d}$ results in 40 classes; in each class, 100 instances are generated.

The *small-item-size* instances of Degraeve and Peeters (2003) are generated similarly for $m = 10, 20, 30, 40, 50, 75, 100$, $c = 0.25, 0.5, 0.75, 1$ and $\bar{d} = 10, 50, 100$, except that $R_1, \ldots, R_m \in (0.1, 0.9)$ for the demand distribution. In the *medium-item-size* instances of Degraeve and Peeters (2003), only $\bar{d} = 50$ is used and the widths are uniformly distributed on $[w_{\min}, cW]$, where $w_{\min} = 500, 1000, 1500$. Both cases have 84 data classes, and 20 instances are generated in each class.

The instances of Vanderbeck (1999) comprise 6 classes with $m = 50$, and 20 instances per class. The first three classes are generated like those of Wäscher and Gau (1996) above with $c = 0.25, 0.5, 0.75$ and $\bar{d} = 50$, the next two classes have widths in $[500, 2500]$ and $[500, 5000]$ with $\bar{d} = 50$, and the sixth class has widths in $[500, 5000]$ and $\bar{d} = 100$.

In the BPP instances of Degraeve and Peeters (2003), $m = 500$ or 1000 weights are uniformly distributed in the intervals $[1, 100]$, $[20, 100]$, $[50, 100]$ as in BPPGEN (Schwerin and Wäscher, 1997), and the capacity $W = 100, 120, 150$; identical items are aggregated for the corresponding CSPs. In each of the 18 resulting classes, 20 instances are generated. The modified BPP instances of Degraeve and Peeters (2003) use $m = 500$, the weight intervals $[1, 10000]$, $[2000, 10000]$, $[5000, 10000]$, and the capacity $W = 10000, 12000, 15000$, again with 20 instances per class.

The BINPACK instances from the OR-Library (Beasley, 1990) comprise two categories. The *uniform* category has the capacity $W = 150$, m weights uniformly distributed in the interval $[20, 100]$, and 20 instances generated for each value of $m = 120, 250, 500, 1000$. (The classes with $m = 500, 1000$ also appear in the BPP category of Degraeve and Peeters (2003), but with different instances.) In the *triplet* category, each bin of capacity $W = 1000$ is filled with exactly three items (the first item w' is picked in $[380, 490]$, the second item w'' in $[250, (W - w')/2)$, and the third item equals $W - w' - w''$). There are 20 instances for each value of $m = 60, 120, 249, 501$.

5.2. Implemented variants

Our codes were programmed in Fortran 77 and run on a notebook PC (Pentium M 755 2 GHz, 1.5 GB RAM) under MS Windows XP.

For solving the dual problem (6), we used a general-purpose bundle code that treats subgradients as dense vectors in double precision. A faster code could exploit the fact that

Table 1: Small-item-size instances of Degraeve and Peeters (2003), $int = all$, $\bar{d} = all$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	26.77	15.14	31	0.00	0.01	113	49	70	1	0	0
20	19.95	53.13	32.51	69	0.01	0.04	120	64	64	0	0	0
30	29.91	79.76	51.90	91	0.02	0.22	130	85	57	0	0	1
40	39.85	105.55	70.41	134	0.04	0.36	134	98	53	0	0	0
50	49.75	132.16	90.20	181	0.08	0.66	134	102	55	0	0	0
75	74.36	197.32	141.82	256	0.24	2.00	149	122	43	0	0	0
100	98.92	263.36	183.88	311	0.40	2.81	165	136	34	0	1	0

each subgradient of θ has the form $\nabla\theta_k = d$ or $\nabla\theta_k = d - Np^k$ (see (18)), with a common integer part d and an integer sparse knapsack solution p^k. Ignoring sparsity, our code requires $m \times M$ memory locations for storing up to $M \geq m+3$ subgradients, and additional workspace of order M^2 for solving the QP subproblem (20) with the routine of Kiwiel (1994). We used $M = m + 3$ to test how "minimal" bundle performs.

The bounded KPs arising in column generation and SHP were solved by the modified version of MT1R (cf. §2.2) with the accuracy tolerance $\epsilon_r = 10^{-5}$ (other choices are discussed in §5.6.2); MT1R's tolerance ϵ was set to 10^{-12}. For column generation, we used the relaxed bounds of (9), because the tighter bounds of (8) produced longer computing times. In contrast, SHP employed in (14) the natural bounds given by (8) with d replaced by d''.

Our implementation of the rounding procedure of §3.1 is slower than necessary because the patterns are recovered as $p^j = (d - \nabla\theta_j)/N$, instead of being stored separately.

5.3. Results for the cutting-stock problem

To ease comparisons, we follow closely the presentation of Degraeve and Peeters (2003). Every data class is identified by three parameters: the number of items m, the interval in which the widths are distributed denoted by *int*, and the average demand $\bar{d}$. An indicator "*all*" for any of these parameters means that the reported results are aggregated over all relevant values for that particular parameter. If a parameter is constant for all instances represented in a table, its value is indicated in the table heading.

Our results for the small-item-size instances of Degraeve and Peeters (2003) with $int = all$, $\bar{d} = all$ are reported in Table 1; full details are given in Tables 9–11 in the Online Supplement to this paper on the journal's website. The columns m_{av} and m'_{av} give the average numbers of items and variables in the associated 0–1 knapsack subproblems. The columns i_{av} and i_{mx}

Table 2: Medium-item-size instances of Degraeve and Peeters (2003), $int = all$, $\bar{d} = 50$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.98	23.09	17.52	29	0.00	0.02	54	48	112	0	1	0
20	19.95	45.58	35.05	58	0.01	0.10	68	50	114	0	1	0
30	29.84	68.47	53.71	93	0.02	0.16	73	73	105	0	0	0
40	39.78	90.65	69.94	120	0.03	0.58	70	63	110	0	0	0
50	49.64	113.69	88.76	156	0.06	0.90	74	65	118	1	1	1
75	74.08	169.10	137.04	232	0.37	8.60	82	73	105	0	0	1
100	98.45	226.07	184.45	295	1.43	62.18	73	72	117	0	4	0

report the average and maximum numbers of iterations of the bundle code. The columns t_{av} and t_{mx} give the average and maximum running times in wall-clock seconds. The column n_e lists the numbers of "early" terminations due to discovering that $ez^* = \lceil \underline{\theta}_k \rceil$ for the incumbent z^* delivered by H0 or H1 *before* bundle terminated on its own. Recall that H1 is called after H0, H2 after H1, etc., unless $ez^* = \lceil \underline{\theta}_k \rceil$ occurs earlier. The columns labelled H1 through H4 give the numbers of instances in which the corresponding heuristic found the best primal value ez^* first (for the remaining instances ez^* was found by H0); a zero entry means that heuristic was not called or did not contribute usefully. The final column n_g reports the numbers of instances with a nonzero final *gap* $g := ez^* - \lceil \underline{\theta}_k \rceil$; we stress that the final gaps never exceeded *one* unit in *all* of our instances. The averages, maxima and sums in Table 1 are taken over the 240 instances used for each value of m.

From the entries for n_e, H1 through H4 and n_g in Table 1, we see that early termination occured on between 47% and 69% of problems, H0 and H1 solved between 70% and 85% of problems, H2 solved almost all the remaining problems, H3 and H4 helped in solving 2 problems, and just one out of the 1680 problems was not solved. Note that the best method LR of Degraeve and Peeters (2003) also could not solve one instance within 15 minutes (two instances within 6 minutes), and its FFD-based rounding heuristic solved 91.6% of problems, whereas our "lighter" heuristics H0 through H2 solved 99.8% of problems.

Our results for the medium-item-size instances of Degraeve and Peeters (2003) are presented in Table 2, where each row gives statistics over the 240 instances used for each value of m (see Tables 12 and 13 for more details). Early termination occured on between 22% and 35% of problems, H0 and H1 solved between 49% and 56% of problems, H2 solved almost all the remaining problems, H3 solved one problem, H4 solved 7 problems, and just two out of the 1680 problems were not solved. The rounding heuristic of Degraeve and Peeters (2003)

Table 3: CSP instances of Wäscher and Gau (1996), *int* = *all*, $\bar{d}$ = *all*

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	14.27	35	0.00	0.02	449	134	192	0	0	0
20	19.96	50.46	30.73	61	0.01	8.35	485	240	183	0	2	0
30	29.90	75.72	48.18	105	0.01	0.13	503	281	161	0	1	0
40	39.84	100.10	65.06	123	0.04	3.31	502	313	160	0	2	2
50	49.73	125.22	84.75	171	0.07	0.46	526	341	138	0	4	1
all	29.88	75.37	48.60	171	0.03	8.35	2465	1309	834	0	9	3

Table 4: CSP instances of Vanderbeck (1999), $m = 50$

$\bar{d}$	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
50	[1, 2500]	49.40	185.30	47.40	71	0.03	0.05	20	18	0	0	0	0
50	[1, 5000]	49.65	143.05	114.05	151	0.20	0.34	13	13	7	0	0	0
50	[1, 7500]	49.75	110.00	111.85	144	0.06	0.11	6	5	8	0	0	0
50	[500, 2500]	49.40	166.10	57.05	77	0.03	0.05	14	14	6	0	0	0
50	[500, 5000]	49.70	128.20	103.65	114	0.14	0.27	11	11	9	0	0	0
100	[500, 5000]	49.70	129.25	104.40	131	0.14	0.32	8	8	12	0	0	0

solved 69.9% of problems, whereas H0 through H2 solved 99.4% of problems.

Comparing Tables 1 and 2, we see that the average and maximum solution times are quite similar in the small- and medium-size-item cases for problem sizes m up to 50. However, for $m = 75$ and 100, in the medium-size-item case the average solution times grow significantly, and the maximum solution times jump up, most spectacularly on the instances with width interval [1500, 2500]; see Table 13. This is due to the poor performance of our knapsack solver on these instances. Similar slowdowns on this interval were reported in (Degraeve and Peeters, 2003, Tab. 4a) already for $m = 20$, i.e., even for smaller problems.

To save space, Table 3 presents only aggregate results on the instances of Wäscher and Gau (1996), with each row giving statistics over the 800 instances used for each value of m. Here our main point is that only three out of 4000 (0.075%) problems were not solved. Our "lighter" heuristics H0 through H2 solved 99.7% of problems, whereas the two best (and more complicated) heuristics RSUC and CSTAOPT of Wäscher and Gau (1996) solved 98.0% and 92.7% of problems, respectively (99.6% if they had been applied together). The fairly large maximum solution time in Tab. 3 stemmed from a single knapsack subproblem.

Table 4 gives our results for the 6 data classes of Vanderbeck (1999) with $m = 50$ and 20 instances per row. Since we used the original instances, the results are not identical to those

Table 5: BPP instances of Degraeve and Peeters (2003)

m	W	int	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
500	100	[1, 100]	99.35	167.20	184.10	221	0.06	0.09	12	1	1	0	0	0
		[20, 100]	80.75	116.00	111.50	123	0.02	0.03	10	2	0	0	0	0
		[50, 100]	51.00	52.00	56.60	63	0.00	0.01	15	0	0	0	0	0
	120	[1, 100]	99.65	181.85	37.05	195	0.29	3.79	17	1	0	0	0	0
		[20, 100]	80.85	131.20	132.80	146	0.03	0.04	14	6	0	0	0	0
		[50, 100]	51.00	62.00	56.55	61	0.00	0.01	13	0	0	0	0	0
	150	[1, 100]	99.45	201.55	1.00	1	0.00	0.00	20	0	0	0	0	0
		[20, 100]	80.85	151.65	86.55	102	0.01	0.02	14	14	5	0	1	0
		[50, 100]	51.00	77.00	64.80	72	0.01	0.01	12	0	0	0	0	0
1000	100	[1, 100]	100.00	183.65	199.20	230	0.07	0.11	12	1	1	0	0	0
		[20, 100]	81.00	117.95	114.25	133	0.02	0.02	14	4	1	0	0	0
		[50, 100]	51.00	52.00	57.35	64	0.00	0.01	9	0	0	0	0	0
	120	[1, 100]	100.00	202.20	25.00	181	0.01	0.04	20	3	0	0	0	0
		[20, 100]	81.00	132.95	143.40	167	0.03	0.04	10	3	2	0	0	0
		[50, 100]	51.00	62.00	56.90	62	0.00	0.01	11	0	0	0	0	0
	150	[1, 100]	100.00	226.15	7.00	121	0.00	0.03	20	1	0	0	0	0
		[20, 100]	81.00	154.90	86.85	101	0.01	0.02	11	11	9	0	0	0
		[50, 100]	51.00	77.00	67.25	77	0.01	0.01	10	0	0	0	0	0

in Tabs. 9 and 13, but the performance of H0 through H2 is similar; in fact H0 through H2 suffice for solving *all* the CSP instances used by Vanderbeck (1999).

Quite suprisingly, all the industrial instances we could find in the literature turned out to be easy for our method: they were solved in a fraction of a second (see Tables 14–16).

5.4. Results for the bin-packing problem

Following Degraeve and Peeters (2003), in the next three tables we present our results for the BPP. Table 5 gives our results for the BPP instances of Degraeve and Peeters (2003) (20 instances per row). All the 360 instances were solved (H4 helped once).

Table 6 reports results for the BINPACK instances from the OR-Library (Beasley, 1990) (20 instances per row). The first four uniform classes were solved by calling H4 just once. However, only 19 out of the 80 triplet instances were solved (with H4 helping on one instance). The remaining instances had unit gaps; the "gap" column gives averages of *relative* gaps $(ez^* - \lceil \underline{\theta}_k \rceil)/\lceil \underline{\theta}_k \rceil$. We add that for the CSP instances of §5.3, the running times of H4 were not excessive, and H4 was called quite infrequently anyway. In contrast, on the triplet classes t249 and t501, the use of H4 increased the running times substantially, as illustrated in Table 7 (the influence of H3 could be ignored). Note that the triplet classes are quite difficult for traditional LP relaxation (Degraeve and Peeters, 2003, Tab. 12).

Table 6: BINPACK uniform and triplet instances

name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	gap	n_g
u120	63.20	88.75	48.60	89	0.00	0.01	20	14	0	0	0	0.0%	0
u250	77.25	129.00	86.40	122	0.01	0.03	19	19	1	0	0	0.0%	0
u500	80.80	151.05	85.90	113	0.01	0.04	16	16	3	0	1	0.0%	0
u1000	81.00	155.00	86.30	97	0.01	0.02	12	12	8	0	0	0.0%	0
t60	49.95	58.80	40.20	56	0.01	0.04	0	1	19	0	0	1.5%	6
t120	86.15	110.75	72.70	91	0.06	0.09	0	1	18	0	1	2.0%	16
t249	140.10	199.15	126.70	146	0.26	0.37	0	1	19	0	0	1.2%	20
t501	194.25	315.40	167.40	189	0.67	1.14	0	0	20	0	0	0.6%	19

Table 7: BINPACK triplet instances without H3 and H4

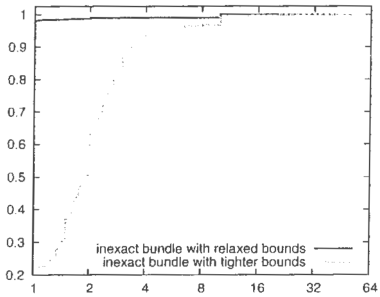
name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	gap	n_g
t60	49.95	58.80	40.20	56	0.00	0.01	0	1	19	1.5%	6
t120	86.15	110.75	72.70	91	0.01	0.02	0	1	19	2.1%	17
t249	140.10	199.15	126.70	146	0.04	0.06	0	1	19	1.2%	20
t501	194.25	315.40	167.40	189	0.08	0.10	0	0	20	0.6%	19

Table 8: Modified BPP instances of Degraeve and Peeters (2003)

W	int	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10000	$[1, 10000]$	488.65	494.05	1484.40	1737	34.95	48.35	14	3	0	0	0	0
	$[2000, 10000]$	485.15	490.20	800.70	916	7.05	9.87	15	1	0	0	0	0
	$[5000, 10000]$	474.75	474.80	457.70	480	1.15	1.35	16	0	0	0	0	0
12000	$[1, 10000]$	486.95	494.55	817.90	1732	25.89	58.02	18	7	1	0	0	0
	$[2000, 10000]$	484.75	492.20	1157.90	1328	15.00	21.33	18	2	0	0	0	0
	$[5000, 10000]$	475.95	480.35	520.75	550	2.20	2.64	15	0	0	0	0	0
15000	$[1, 10000]$	487.90	497.15	293.60	1171	8.00	67.00	18	6	0	0	1	1
	$[2000, 10000]$	482.70	494.25	805.05	1144	16.19	29.37	16	16	4	0	0	0
	$[5000, 10000]$	475.25	486.95	691.50	786	5.14	6.31	13	0	0	0	0	0

Table 8 presents our results for the modified BPP classes of Degraeve and Peeters (2003) (20 instances per row as described in §5.1). Just one out of the 180 problems was not solved (H4 helped on one problem). The transformation into a CSP reduced the number of items by at most 5% on average. For almost 500 variables, the large iteration numbers and running times are not too suprising.

Figure 1: Performance profile for inexact bundle with tight vs. relaxed bounds



5.5. Impact of tighter knapsack bounds

The results of §5.3 were obtained for the relaxed bounds of (9). Using the tighter bounds of (8) allowed us to solve just two more instances at the expense of longer running times (see Tabs. 17–19). To save space, from now on we employ the *standard* set of the 7360 instances from Tabs. 1–3 to evaluate our heuristics, and its *reduced* subset with $m \geq 30$ (4800 instances) for performance profiles (Dolan and Moré, 2002), with zero running times replaced by 0.001 due to the poor resolution of our timer. The performance profile of tighter vs. relaxed bounds is given in Fig. 1; it plots the portion of instances $\rho_s(\tau)$ on which a particular variant was not slower than the fastest variant by more than a given ratio τ.

5.6. Impact of evaluation errors

5.6.1. Comparison with exact bundle

When the dual objective evaluations happen to be exact, our code runs essentially like the standard bundle of Feltenmark and Kiwiel (2000). Figure 2 gives the performance profile of inexact bundle ($\epsilon_r = 10^{-5}$) with relaxed bounds vs. exact bundle ($\epsilon_r = 0$) with relaxed or tighter bounds. Referring to Tabs. 22–27 for details, we only note that the running times increased quite dramatically (usually at least 30 times) in the exact case, although the iteration numbers and the performance of our heuristics did not change significantly.

Figure 2: Performance profile for inexact bundle with relaxed bounds vs. exact bundle with tight/relaxed bounds

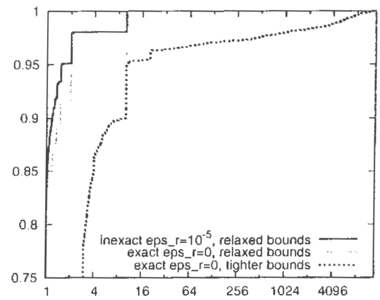
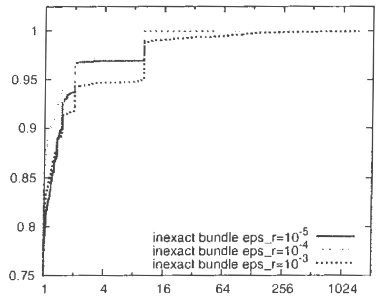


Figure 3: Performance profile for relative error tolerances

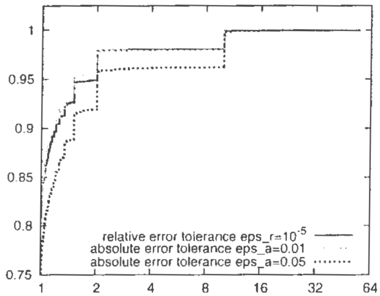


5.6.2. Other choices of the relative error tolerance

In the initial version of this paper we used the accuracy tolerance $\epsilon_r = 10^{-8}$; the results were very close to those in Tabs. 1–18 (where $\epsilon_r = 10^{-5}$). Figure 3 gives the performance profile for $\epsilon_r = 10^{-5}$, 10^{-4} and 10^{-3} (see also Tabs. 28–33). Here $\epsilon_r = 10^{-4}$ did not improve on our standard choice of $\epsilon_r = 10^{-5}$ (giving one more gap in Tab. 28), whereas $\epsilon_r = 10^{-3}$ was too large, causing our heuristics to fail more frequently (168 more gaps in Tabs. 31–33).

Further insight may be gained as follows. By (10), the absolute error in evaluating θ is bounded by $N\epsilon_r$ once $\bar{\zeta}$ gets close to 1. The upper bound $N := ez^*$ delivered by FFD (cf. §3.3) is usually close to the optimal primal value v_*. Typical instances have the *integer*

Figure 4: Performance profile for absolute error tolerances



round-up property $\lceil \theta_* \rceil = v_*$, but our heuristics fail if we can't find a lower bound $\underline{\theta}_k > v_* - 1$. Thus we may expect failures when the absolute errors get close to $N\epsilon_r > 1$. Now, in Tables 31–33 the average values of v_* and N grow linearly with m, reaching order 5000, 2875 and 1250 for the final classes, where $N\epsilon_r > 1$ for $\epsilon_r = 10^{-3}$; thus the small percentage of failures suggests that the actual errors tended to be smaller than their upper bounds.

5.6.3. Absolute error tolerances

In view of the discussion in §5.6.2, we also considered choosing ϵ_r so that the evaluation errors did not exceed a given *absolute error tolerance* $\epsilon_a < 1$ (with SHP using $\epsilon_r = 10^{-5}$ as in §5.3). Specifically, for evaluating θ we used $\epsilon_r := \epsilon_a/N$. Figure 4 gives the performance profile for $\epsilon_a = 0.01$ and 0.05 vs. the standard $\epsilon_r = 10^{-5}$ (see also Tabs. 34–39). Our results for $\epsilon_a = 0.01$ were very close to those for $\epsilon_r = 10^{-5}$, whereas $\epsilon_a = 0.05$ was too large, causing our heuristics to fail more frequently (16 more gaps in Tabs. 37–39).

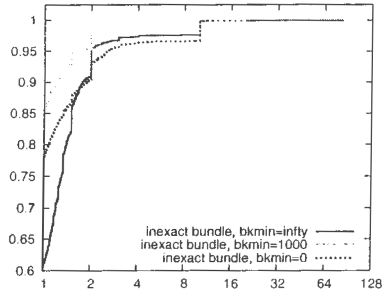
5.6.4. More inexact null steps

We now consider a modification in which our KP solver exits once at least `bkmin` backtrackings have occured, for a given parameter `bkmin`, and the incumbent value ζ satisfies

$$\zeta > 1 + \left(u^{k+1}d - \theta^k_{\hat{u}} - \kappa v_k\right)/N, \tag{26}$$

so that $\zeta_{k+1} := \zeta$ yields a null step; cf. (22) (normally $u^{k+1}d > \theta^k_{\hat{u}} + \kappa v_k$ and (26) holds iff (22) fails). Such "more inexact" null steps may save KP work, but shallower cuts may

Figure 5: Performance profile for bkmin



yield slower convergence; see (Kiwiel, 2006b, §4.2) for a general discussion of relaxed null steps. Figure 5 gives the performance profile for bkmin $= 0$, 1000 and ∞ with $\epsilon_r = 10^{-5}$ (see also Tabs. 40–45). Relative to the standard bkmin $= \infty$, for bkmin $= 0$ the average iteration numbers grew by 59–114% on the largest instances, and four more gaps occured. In contrast, for bkmin $= 1000$ the average iteration numbers grew by only 5–13% on the largest instances, the solution times decreased noticeably, and three gaps disappeared. On the other hand, the maximum iteration numbers increased substantially on the larger instances, giving some cause for concern.

5.6.5. A discussion of error tolerances

Although in general one may expect tradeoffs between the accuracy of subproblem solutions and the speed of convergence, for the CSP such tradeoffs may have little practical impact, since Tables 9–30 exhibit fairly small variations in iteration numbers and computing times for "reasonable" accuracy tolerances. Therefore, we would not expect much gain from *dynamic tolerance adjustment*: loose at the beginning and progressively decreasing.

We add that dynamic handling of the accuracy may be important in general, especially if the oracle's work depends "continuously" on the accuracy required. However, this need not be the case for our MT1R, which seems to have the following properties: (1) its work explodes on some subproblems when the accuracy required is "too high"; and (2) its work does not vary much otherwise. Thus the main point is to avoid accuracies that are "too high", or "too low" for the dual solver to succeed, whereas for all "intermediate" accuracies, the solution time should not vary significantly (unless smaller accuracies affect the iteration numbers "more than proportionally"). We conjecture that similar effects are likely to hold for other integer-programming applications with branch-and-bound oracles that deliver relatively good incumbents quickly.

5.7. Impact of various heuristics

For the 7,538 CSP instances reported in Tabs. 1–4 and 14–16, our heuristics H3 and H4 helped in solving 3 and 21 problems, respectively, and 6 problems were not solved. When H3 was switched off, H4 solved the three instances previously solved by H3, with the same timings. Thus H3 could be omitted, but it might become more useful on other instances. On the other hand, it is worth observing that when both H3 and H4 were switched off, our "lighter" heuristics H1 and H2 performed quite well, solving 99.64% of problems.

In an attempt to assess the importance of the combination of oversupply reduction (Step 2 of Procedure 1), rounding up (Step 3), and residual problem extension (Step 5), we tested a version of the residual rounding heuristic named H5 that simply rounds the final relaxed primal solution down, and performs FFD on the residual problem to augment the rounded down solution. With Steps 2, 3, and 5 of the rounding procedure omitted, this heuristic H5 was able to optimally solve only 87.01% of the standard instances, as opposed to 99.64% for the default implementation of H0, H1 and H2 (see Tabs. 46–48). Thus these steps (in tandem) are *very important* to its overall success.

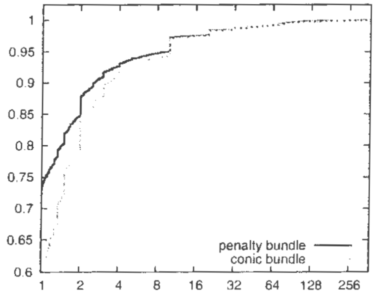
Our next improvement on H5, named H6, consists in calling Procedure 1 with only Step 2 omitted, and Step 4 using FFD. H6 performs much better than H5, solving 96.00% of problems (see Tabs. 49–51). Thus the rounding procedure of Belov and Scheithauer (2002) may yield significant improvements also for FFD.

Finally, we note that H2 and H4 improve on H6 by using Step 2 of Procedure 1 and either SHP or SVC in addition to FFD at Step 4. Specifically, H1 and H2 solved 99.64% of problems, and together with H4 they solved 99.92% of problems. To save space, the results for H2 alone are omitted.

5.8. Comparisons with other procedures from the literature

In view of (5)–(6), our algorithm may be regarded as an exact *penalty* method for the constrained problem of maximizing ud s.t. $up(u) \leq 1$, $u \geq 0$. This problem can also be

Figure 6: Performance profile for conic vs. penalty bundle



solved by the *conic* variant of Kiwiel and Lemaréchal (2007). Figure 6 gives the performance profile for the conic vs. penalty variant. The conic variant was slightly slower, and gave one more gap on the standard set; cf. Tabs. 52–54.

The comparison in (Kiwiel and Lemaréchal, 2007, §7.4) of the conic variant with the procedures of Degraeve and Peeters (2003) in terms of the numbers of oracle calls carries over to the penalty variant as well, since both variants behaved similarly. Although proper timing comparisons are not available, Table 55 in the supplement suggests that our code may compete with the procedures of Degraeve and Peeters (2003), at least on some instances.

Finally, we add that Table 60 in the supplement shows that our standard variant (with $\epsilon_r = 10^{-5}$ and relaxed bounds) is much faster than the algorithms tested in (Briant et al., 2007, §2.2), with speedups of at least 8 for the smallest instances, and of order 11–90 for the larger instances.

6. Conclusions

For cutting-stock problems, we have shown that an inexact bundle approach to solving the LP relaxation, coupled with rounding heuristics, is a method that is able to effectively solve many cutting-stock instances from the literature. By solving the KP subproblems only to a relative accuracy of $\epsilon_r = 10^{-5}$ we get (almost uniformly) speedup of the order of at least 30 in average on larger instances. Although our heuristics combine several well-known ideas from the literature, our two "lighter" heuristics H1 and H2 performed suprisingly well, solving

99.64% of standard test problems, and together with our "heavier" heuristic H4 they solved 99.92% of problems.

Acknowledgments

I would like to thank the Associate Editor and the two anonymous referees for helpful comments. Further, I am grateful to G. Belov, Z. Degraeve, M. Peeters, D. Pisinger, G. Scheithauer and L. Schrage for extensive discussions, and F. Vanderbeck and G. Wäscher for sharing their instances. Special thanks go to C. Lemaréchal for inspiring this work. This research was supported by the INRIA New Investigation Grant "Convex Optimization and Dantzig-Wolfe Decomposition".

References

Beasley, J. E. 1990. OR-Library: Distributing test problems by electronic mail. *J. Oper. Res. Soc.* **41** 1069–1072.

Belov, G., G. Scheithauer. 2002. A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. *European J. Oper. Res.* **141** 274–294.

Belov, G., G. Scheithauer. 2006. A branch-and-cut-and-price algorithm for one- and two-dimensional two-stage cutting problems. *European J. Oper. Res.* **171** 85–106.

Belov, G., G. Scheithauer. 2007. Setup and open stacks minimization in one-dimensional stock cutting. *INFORMS J. Comput.* **19** 27–35.

Ben Amor, H., J. Desrosiers, A. Frangioni. 2004. Stabilization in column generation. Tech. Rep. G-2004-62, GERAD, Montreal.

Ben Amor, H., J. M. Valério de Carvalho. 2005. Cutting stock problems. G. Desaulniers, J. Desrosiers, M. M. Salomon, eds., *Column Generation*. Springer US, New York, 131–161.

Briant, O., C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, F. Vanderbeck. 2007. Comparison of bundle and classical column generation. *Math. Programming* **113** 299–344.

Chvátal, V. 1983. *Linear Programming*. Freeman, New York, N.Y.

Degraeve, Z., M. Peeters. 2003. Optimal integer solutions to industrial cutting stock problems: Part 2: Benchmark results. *INFORMS J. Comput.* **15** 58–81.

Degraeve, Z., L. Schrage. 1999. Optimal integer solutions to industrial cutting stock problems. *INFORMS J. Comput.* **11** 406–419.

Dolan, E. D., J. J. Moré. 2002. Benchmarking optimization software with performance profiles. *Math. Programming* **91** 201–213.

Feltenmark, S., K. C. Kiwiel. 2000. Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems. *SIAM J. Optim.* **10** 697–721.

Gau, T., G. Wäscher. 1995. CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem. *European J. Oper. Res.* **84** 572–579.

Gilmore, P. C., R. E. Gomory. 1961. A linear programming approach to the cutting-stock problem. *Oper. Res.* **9** 849–859.

Holthaus, O. 2002. Decomposition approaches for solving the integer one-dimensional cutting stock problem with different types of standard lengths. *European J. Oper. Res.* **141** 295–312.

Kellerer, Hans, Ulrich Pferschy, David Pisinger. 2004. *Knapsack Problems*. Springer, Berlin.

Kiwiel, K. C. 1994. A Cholesky dual method for proximal piecewise linear programming. *Numer. Math.* **68** 325–340.

Kiwiel, K. C. 2006a. A proximal bundle method with approximate subgradient linearizations. *SIAM J. Optim.* **16** 1007–1023.

Kiwiel, K. C. 2006b. A proximal-projection bundle method for Lagrangian relaxation, including semidefinite programming. *SIAM J. Optim.* **17** 1015–1034.

Kiwiel, K. C., C. Lemaréchal. 2007. An inexact bundle variant suited to column generation. *Math. Programming* **?** DOI 10.1007/s10107-007-0187-4.

Martello, S., P. Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, New York.

Nitsche, C., G. Scheithauer, J. Terno. 1999. Tighter relaxations for the cutting stock problem. *European J. Oper. Res.* **112** 654–663.

Scheithauer, G., J. Terno, A. Müller, G. Belov. 2001. Solving one-dimensional cutting stock problems exactly using a cutting plane algorithm. *J. Oper. Res. Soc.* **52** 1390–1401.

Schwerin, P., G. Wäscher. 1997. The bin-packing problem: A problem generator and some numerical experiments. *Int. Trans. Oper. Res.* **4** 337–389.

Stadtler, H. 1990. One-dimensional cutting stock problem in the aluminium industry and its solution. *European J. Oper. Res.* **44** 209–223.

Valério de Carvalho, J. M. 2005. Using extra dual cuts to accelerate column generation. *INFORMS J. Comput.* **17** 175–182.

Vance, P. H. 1998. Branch and price algorithms for the one-dimensional cutting stock problem. *Comput. Optim. Appl.* **9** 212–228.

Vanderbeck, F. 1999. Computational study of a column generation algorithm for bin packing and cutting stock problems. *Math. Programming* **86** 565–594.

Vanderbeck, F. 2002. Extending Dantzig's bound to the bounded multiple-class binary knapsack problem. *Math. Programming* **94** 125–136.

Wäscher, G., T. Gau. 1996. Heuristics for the integer one-dimensional cutting stock problem. *OR Spectrum* **18** 131–144.

Online Supplement for

An Inexact Bundle Approach to Cutting-Stock Problems

INFORMS Journal on Computing

Krzysztof C. Kiwiel
Systems Research Institute, Newelska 6, 01–447 Warsaw, Poland, kiwiel@ibspan.waw.pl

A. Additional tables

A.1. Results for the cutting-stock problem

Tables 9–11 below give details for the small-item-size instances of Degraeve and Peeters (2003). The averages, maxima and sums in Table 9 are taken over 20 instances for each interval, and thus over 80 instances for each "*all*" row. In Table 10, there are 60 instances per interval (i.e., 20 instances for each value of the average demand $\bar{d} = 10, 50, 100$), and each "*all*" row gives statistics over the 240 instances used for each value of m. Finally, each row in Table 11 reports statistics over 80 instances (obtained from the 20 instances used for each of the four width intervals).

Our detailed results for the medium-item-size instances of Degraeve and Peeters (2003) are presented in Tables 12 and 13, where each "*all*" row gives statistics over the 240 instances used for each value of m.

Tables 14–16 give our results for the industrial instances of Vance (1998) (as numbered in (Degraeve and Peeters, 2003, Tab. 7)), (Vanderbeck, 1999, Tab. 1) and Degraeve and Schrage (1999) (as named in (Degraeve and Peeters, 2003, Tab. 9)). The final column identifies the heuristic which delivered the optimal solution; in other words, H0 through H2 solved all these instances except for a single instance solved by H3.

Table 9: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	[1, 2500]	10.00	38.15	7.95	20	0.00	0.01	19	13	0	0	0	0
	[1, 5000]	10.00	29.45	19.75	31	0.00	0.01	5	7	13	0	0	0
	[1, 7500]	9.95	22.35	19.95	29	0.00	0.01	0	0	8	0	0	0
	[1, 10000]	10.00	19.95	18.10	24	0.00	0.01	3	0	6	1	0	0
	all	9.99	27.47	16.44	31	0.00	0.01	27	20	27	1	0	0
20	[1, 2500]	20.00	78.65	13.00	21	0.00	0.01	20	12	0	0	0	0
	[1, 5000]	19.95	57.10	42.10	56	0.01	0.04	7	7	12	0	0	0
	[1, 7500]	20.00	45.15	42.10	59	0.01	0.01	3	2	8	0	0	0
	[1, 10000]	20.00	38.95	36.45	52	0.00	0.01	4	1	5	0	0	0
	all	19.99	54.96	33.41	59	0.01	0.04	34	22	25	0	0	0
30	[1, 2500]	29.90	116.85	26.40	51	0.01	0.03	20	15	0	0	0	0
	[1, 5000]	29.90	87.30	69.25	91	0.04	0.08	9	9	11	0	0	0
	[1, 7500]	30.00	68.50	65.40	91	0.01	0.02	5	3	8	0	0	0
	[1, 10000]	29.95	60.25	58.15	69	0.01	0.02	4	2	3	0	0	0
	all	29.94	83.22	54.80	91	0.02	0.08	38	29	22	0	0	0
40	[1, 2500]	39.80	153.20	39.65	76	0.02	0.07	19	17	1	0	0	0
	[1, 5000]	39.85	113.20	92.50	121	0.12	0.21	14	14	6	0	0	0
	[1, 7500]	39.90	89.20	92.35	121	0.03	0.05	6	6	6	0	0	0
	[1, 10000]	39.90	76.15	78.25	108	0.02	0.02	3	1	5	0	0	0
	all	39.86	107.94	75.69	121	0.05	0.21	42	38	18	0	0	0
50	[1, 2500]	49.60	190.70	35.75	51	0.02	0.04	20	14	0	0	0	0
	[1, 5000]	49.70	145.30	116.55	171	0.18	0.42	16	16	4	0	0	0
	[1, 7500]	49.75	113.30	124.85	151	0.07	0.13	5	3	13	0	0	0
	[1, 10000]	50.00	99.95	105.30	131	0.04	0.07	1	2	3	0	0	0
	all	49.76	137.31	95.61	171	0.08	0.42	42	35	20	0	0	0
75	[1, 2500]	73.85	285.85	71.75	115	0.07	0.15	19	17	1	0	0	0
	[1, 5000]	74.10	218.00	167.40	256	0.41	1.33	18	18	2	0	0	0
	[1, 7500]	74.80	170.10	196.40	226	0.29	0.61	5	5	7	0	0	0
	[1, 10000]	74.75	145.15	158.75	218	0.12	0.23	3	2	4	0	0	0
	all	74.38	204.78	148.57	256	0.22	1.33	45	42	14	0	0	0
100	[1, 2500]	98.50	374.00	101.50	120	0.13	0.21	20	20	0	0	0	0
	[1, 5000]	99.05	286.45	183.95	261	0.45	1.95	17	17	3	0	0	0
	[1, 7500]	99.30	227.35	272.70	311	0.78	1.27	14	13	5	0	0	0
	[1, 10000]	99.45	194.55	224.65	294	0.31	0.65	5	3	3	0	0	0
	all	99.08	270.59	195.70	311	0.42	1.95	56	53	11	0	0	0

Table 10: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d}$ = *all*

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	[1, 2500]	10.00	36.70	8.02	28	0.00	0.01	54	30	5	0	0	0
	[1, 5000]	9.98	28.80	16.13	31	0.00	0.01	22	12	33	0	0	0
	[1, 7500]	9.98	22.10	18.17	29	0.00	0.01	19	4	18	0	0	0
	[1, 10000]	10.00	19.48	18.25	31	0.00	0.01	18	3	14	1	0	0
	all	9.99	26.77	15.14	31	0.00	0.01	113	49	70	1	0	0
20	[1, 2500]	19.95	74.15	14.92	61	0.00	0.02	58	31	2	0	0	0
	[1, 5000]	19.90	56.25	37.57	56	0.01	0.04	30	25	29	0	0	0
	[1, 7500]	19.97	43.53	41.33	69	0.00	0.01	16	6	21	0	0	0
	[1, 10000]	20.00	38.57	36.23	52	0.00	0.01	16	2	12	0	0	0
	all	19.95	53.13	32.51	69	0.01	0.04	120	64	64	0	0	0
30	[1, 2500]	29.85	110.18	22.02	51	0.01	0.03	59	36	1	0	0	0
	[1, 5000]	29.88	84.18	62.42	91	0.04	0.22	35	33	24	0	0	1
	[1, 7500]	29.95	66.50	65.53	91	0.01	0.03	16	10	19	0	0	0
	[1, 10000]	29.95	58.18	57.65	83	0.01	0.02	20	6	13	0	0	0
	all	29.91	79.76	51.90	91	0.02	0.22	130	85	57	0	0	1
40	[1, 2500]	39.75	146.80	31.67	76	0.02	0.07	56	36	4	0	0	0
	[1, 5000]	39.87	111.88	80.00	134	0.09	0.36	44	42	16	0	0	0
	[1, 7500]	39.92	88.75	93.35	121	0.03	0.10	20	13	19	0	0	0
	[1, 10000]	39.87	74.78	76.62	108	0.02	0.03	14	7	14	0	0	0
	all	39.85	105.55	70.41	134	0.04	0.36	134	98	53	0	0	0
50	[1, 2500]	49.60	182.70	32.93	71	0.02	0.06	60	37	0	0	0	0
	[1, 5000]	49.65	140.88	107.05	181	0.20	0.66	41	40	19	0	0	0
	[1, 7500]	49.82	110.80	122.42	151	0.07	0.14	19	16	24	0	0	0
	[1, 10000]	49.93	94.27	98.40	131	0.03	0.07	14	9	12	0	0	0
	all	49.75	132.16	90.20	181	0.08	0.66	134	102	55	0	0	0
75	[1, 2500]	73.82	271.55	57.58	115	0.06	0.15	59	43	1	0	0	0
	[1, 5000]	74.23	209.83	158.07	256	0.53	2.00	49	49	11	0	0	0
	[1, 7500]	74.67	165.52	190.80	239	0.26	0.61	31	26	16	0	0	0
	[1, 10000]	74.72	142.38	160.85	227	0.12	0.27	10	4	15	0	0	0
	all	74.36	197.32	141.82	256	0.24	2.00	149	122	43	0	0	0
100	[1, 2500]	98.13	359.88	75.18	174	0.10	0.21	58	41	2	0	0	0
	[1, 5000]	98.90	280.47	167.22	277	0.44	2.81	53	50	7	0	0	0
	[1, 7500]	99.18	221.73	268.80	311	0.76	1.54	37	35	14	0	1	0
	[1, 10000]	99.45	191.37	224.30	294	0.29	0.65	17	10	11	0	0	0
	all	98.92	263.36	183.88	311	0.40	2.81	165	136	34	0	1	0

Table 11: Small-item-size instances of Degraeve and Peeters (2003), *int* = *all*

m	$\bar{d}$	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	10	10.00	24.95	10.77	26	0.00	0.01	69	14	6	0	0	0
	50	9.99	27.47	16.44	31	0.00	0.01	27	20	27	1	0	0
	100	9.99	27.89	18.21	31	0.00	0.01	17	15	37	0	0	0
20	10	19.94	48.60	25.93	58	0.00	0.02	57	16	9	0	0	0
	50	19.99	54.96	33.41	59	0.01	0.04	34	22	25	0	0	0
	100	19.94	55.81	38.20	69	0.01	0.03	29	26	30	0	0	0
30	10	29.91	73.34	43.33	91	0.01	0.22	57	28	5	0	0	1
	50	29.94	83.22	54.80	91	0.02	0.08	38	29	22	0	0	0
	100	29.88	82.73	57.59	91	0.02	0.12	35	28	30	0	0	0
40	10	39.83	96.79	56.81	109	0.02	0.11	59	29	9	0	0	0
	50	39.86	107.94	75.69	121	0.05	0.21	42	38	18	0	0	0
	100	39.86	111.94	78.72	134	0.05	0.36	33	31	26	0	0	0
50	10	49.70	121.30	75.41	151	0.05	0.36	59	35	6	0	0	0
	50	49.76	137.31	95.61	171	0.08	0.42	42	35	20	0	0	0
	100	49.79	137.88	99.58	181	0.11	0.66	33	32	29	0	0	0
75	10	74.41	181.36	121.49	216	0.18	1.15	61	41	7	0	0	0
	50	74.38	204.78	148.57	256	0.22	1.33	45	42	14	0	0	0
	100	74.29	205.83	155.41	239	0.32	2.00	43	39	22	0	0	0
100	10	98.90	243.64	154.55	310	0.28	1.69	63	36	3	0	0	0
	50	99.08	270.59	195.70	311	0.42	1.95	56	53	11	0	0	0
	100	98.78	275.86	201.38	303	0.49	2.81	46	47	20	0	1	0

Table 12: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	[500, 2500]	10.00	33.80	13.65	23	0.00	0.02	13	13	7	0	0	0
	[1000, 2500]	10.00	31.40	15.45	25	0.00	0.01	9	9	11	0	0	0
	[1500, 2500]	9.90	29.70	16.40	25	0.00	0.01	4	4	16	0	0	0
	[500, 5000]	10.00	28.15	19.10	25	0.00	0.01	5	5	13	0	1	0
	[1000, 5000]	10.00	24.70	17.30	22	0.00	0.01	5	7	12	0	0	0
	[1500, 5000]	9.90	22.50	17.40	23	0.00	0.01	2	5	15	0	0	0
	[500, 7500]	10.00	20.35	20.00	29	0.00	0.01	3	1	10	0	0	0
	[1000, 7500]	10.00	19.00	18.65	24	0.00	0.01	1	0	8	0	0	0
	[1500, 7500]	10.00	17.40	19.25	25	0.00	0.01	0	0	8	0	0	0
	[500, 10000]	10.00	18.30	18.95	28	0.00	0.00	5	0	3	0	0	0
	[1000, 10000]	10.00	16.50	17.35	22	0.00	0.01	3	2	4	0	0	0
	[1500, 10000]	10.00	15.30	16.75	20	0.00	0.01	4	2	5	0	0	0
	all	9.98	23.09	17.52	29	0.00	0.02	54	48	112	0	1	0
20	[500, 2500]	20.00	67.00	26.85	58	0.01	0.08	16	16	4	0	0	0
	[1000, 2500]	19.95	63.30	33.95	58	0.02	0.10	13	13	7	0	0	0
	[1500, 2500]	19.75	59.25	39.00	49	0.02	0.03	10	10	10	0	0	0
	[500, 5000]	19.95	52.40	39.85	45	0.01	0.02	3	3	17	0	0	0
	[1000, 5000]	19.95	48.30	36.35	41	0.01	0.02	2	2	18	0	0	0
	[1500, 5000]	19.95	45.75	34.50	41	0.00	0.01	1	2	18	0	0	0
	[500, 7500]	19.95	40.50	39.70	52	0.00	0.01	3	1	7	0	0	0
	[1000, 7500]	20.00	37.55	37.90	44	0.00	0.01	4	2	10	0	0	0
	[1500, 7500]	20.00	34.30	34.05	41	0.00	0.01	3	1	8	0	1	0
	[500, 10000]	20.00	34.80	33.90	48	0.00	0.01	5	0	4	0	0	0
	[1000, 10000]	19.95	32.40	32.65	45	0.00	0.01	4	0	5	0	0	0
	[1500, 10000]	20.00	31.35	31.95	40	0.00	0.01	4	0	6	0	0	0
	all	19.95	45.58	35.05	58	0.01	0.10	68	50	114	0	1	0
30	[500, 2500]	29.65	100.25	39.35	51	0.01	0.03	16	16	4	0	0	0
	[1000, 2500]	29.85	95.30	40.65	55	0.01	0.03	11	11	9	0	0	0
	[1500, 2500]	29.50	88.50	59.05	93	0.06	0.16	10	10	10	0	0	0
	[500, 5000]	30.00	79.65	61.00	71	0.03	0.06	6	6	14	0	0	0
	[1000, 5000]	29.80	72.95	57.75	71	0.02	0.11	9	9	11	0	0	0
	[1500, 5000]	29.60	66.70	53.80	65	0.01	0.02	6	8	12	0	0	0
	[500, 7500]	29.95	62.25	63.90	75	0.01	0.02	4	3	7	0	0	0
	[1000, 7500]	30.00	55.60	58.95	71	0.01	0.02	3	2	10	0	0	0
	[1500, 7500]	29.90	51.75	55.85	69	0.01	0.01	1	1	10	0	0	0
	[500, 10000]	29.95	52.30	52.00	67	0.00	0.01	2	2	6	0	0	0
	[1000, 10000]	29.95	49.85	53.05	64	0.00	0.01	3	3	7	0	0	0
	[1500, 10000]	29.95	46.55	49.15	61	0.00	0.01	2	2	5	0	0	0
	all	29.84	68.47	53.71	93	0.02	0.16	73	73	105	0	0	0
40	[500, 2500]	39.75	135.85	46.45	61	0.02	0.04	15	15	5	0	0	0
	[1000, 2500]	39.65	125.35	50.60	68	0.02	0.04	12	12	8	0	0	0
	[1500, 2500]	39.30	117.90	66.90	101	0.15	0.58	9	9	11	0	0	0
	[500, 5000]	39.85	103.30	82.05	101	0.07	0.12	9	9	11	0	0	0
	[1000, 5000]	39.80	95.50	76.10	99	0.05	0.09	6	7	13	0	0	0
	[1500, 5000]	39.75	90.85	71.70	84	0.03	0.06	4	4	16	0	0	0
	[500, 7500]	39.80	81.05	88.40	120	0.02	0.04	1	0	11	0	0	0
	[1000, 7500]	39.90	73.90	81.00	96	0.02	0.03	3	2	9	0	0	0
	[1500, 7500]	39.90	71.35	76.05	89	0.01	0.03	3	2	10	0	0	0
	[500, 10000]	39.85	68.80	71.15	93	0.01	0.02	2	2	6	0	0	0
	[1000, 10000]	39.90	63.50	65.55	80	0.01	0.02	2	0	5	0	0	0
	[1500, 10000]	39.90	60.40	63.30	82	0.01	0.02	4	1	5	0	0	0
	all	39.78	90.65	69.94	120	0.03	0.58	70	63	110	0	0	0

Table 13: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
50	[500, 2500]	49.60	170.70	53.60	71	0.02	0.04	20	20	0	0	0	0
	[1000, 2500]	49.20	155.95	64.55	83	0.03	0.05	8	8	12	0	0	0
	[1500, 2500]	49.00	147.00	75.00	109	0.24	0.90	2	3	17	0	0	0
	[500, 5000]	49.45	127.80	104.05	128	0.15	0.26	10	10	10	0	0	0
	[1000, 5000]	49.80	121.15	92.75	108	0.08	0.15	8	8	12	0	0	0
	[1500, 5000]	49.55	114.20	92.40	123	0.06	0.09	4	4	16	0	0	0
	[500, 7500]	49.75	104.90	120.35	156	0.05	0.11	4	3	9	0	1	0
	[1000, 7500]	49.70	93.70	106.30	125	0.04	0.05	2	2	14	1	0	0
	[1500, 7500]	49.90	86.65	96.20	112	0.03	0.07	4	3	7	0	0	1
	[500, 10000]	49.85	85.30	93.25	118	0.03	0.04	4	2	8	0	0	0
	[1000, 10000]	50.00	83.05	89.65	111	0.02	0.03	4	0	9	0	0	0
	[1500, 10000]	49.90	73.90	77.00	108	0.01	0.03	4	2	4	0	0	0
	all	49.64	113.69	88.76	156	0.06	0.90	74	65	118	1	1	1
75	[500, 2500]	73.65	251.20	83.00	103	0.06	0.08	13	13	7	0	0	0
	[1000, 2500]	73.40	232.30	88.60	101	0.05	0.06	5	5	15	0	0	0
	[1500, 2500]	71.85	215.55	102.20	164	2.73	8.60	3	3	17	0	0	0
	[500, 5000]	74.10	193.55	163.45	216	0.50	0.98	9	9	11	0	0	0
	[1000, 5000]	74.00	181.05	143.95	175	0.29	0.47	12	12	8	0	0	0
	[1500, 5000]	74.15	167.65	141.75	205	0.18	0.41	10	11	9	0	0	1
	[500, 7500]	74.70	154.65	181.65	232	0.20	0.50	7	7	11	0	0	0
	[1000, 7500]	74.50	138.80	171.25	201	0.12	0.16	6	7	5	0	0	0
	[1500, 7500]	74.50	128.95	153.60	178	0.09	0.15	6	3	7	0	0	0
	[500, 10000]	74.70	129.85	146.80	180	0.09	0.13	6	0	3	0	0	0
	[1000, 10000]	74.65	120.65	137.85	166	0.07	0.12	3	2	4	0	0	0
	[1500, 10000]	74.70	114.95	130.40	180	0.05	0.09	2	1	8	0	0	0
	all	74.08	169.10	137.04	232	0.37	8.60	82	73	105	0	0	1
100	[500, 2500]	98.10	336.05	104.70	120	0.08	0.11	13	13	7	0	0	0
	[1000, 2500]	96.30	303.40	106.05	116	0.07	0.10	3	3	17	0	0	0
	[1500, 2500]	95.55	286.65	128.60	210	13.57	62.18	3	3	17	0	0	0
	[500, 5000]	98.70	263.50	205.00	260	0.81	2.06	17	17	3	0	0	0
	[1000, 5000]	98.65	240.15	197.50	220	0.72	1.00	9	9	11	0	0	0
	[1500, 5000]	98.70	225.25	189.35	269	0.48	0.73	8	8	10	0	2	0
	[500, 7500]	99.15	202.85	252.35	295	0.42	0.54	4	6	10	0	0	0
	[1000, 7500]	99.40	188.45	240.60	295	0.31	0.39	8	8	10	0	1	0
	[1500, 7500]	98.70	174.60	212.80	236	0.22	0.32	1	1	14	0	1	0
	[500, 10000]	99.40	174.70	210.00	272	0.23	0.39	1	0	7	0	0	0
	[1000, 10000]	99.45	163.90	192.15	235	0.17	0.43	3	2	4	0	0	0
	[1500, 10000]	99.25	153.35	174.25	208	0.12	0.16	3	2	7	0	0	0
	all	98.45	226.07	184.45	295	1.43	62.18	73	72	117	0	4	0

Table 14: Industrial CSP instances of Vance (1998)

inst.	m	m'	i	t	n_e	Hi
1	2	7	1	0.00	1	H0
2	2	8	3	0.00	1	H1
3	3	11	1	0.00	1	H0
4	5	17	1	0.00	1	H0
5	14	50	15	0.01	1	H1
6	5	19	1	0.00	1	H0
7	4	14	1	0.00	1	H0
8	7	27	10	0.00	0	H2
9	11	46	12	0.00	1	H1
10	3	9	2	0.00	1	H0
11	2	7	1	0.00	1	H0
12	6	23	1	0.00	1	H0
13	2	9	1	0.00	1	H0
14	3	11	4	0.00	1	H1
15	7	20	8	0.00	1	H1
16	4	9	1	0.00	1	H0
17	12	42	24	0.00	0	H2
18	14	44	15	0.00	1	H1
19	5	15	13	0.01	0	H2
20	11	31	21	0.00	0	H2
21	9	27	16	0.00	0	H2
22	8	25	16	0.00	0	H2
23	7	20	8	0.00	1	H1
24	7	22	13	0.00	0	H2
25	12	39	13	0.01	1	H1
26	6	18	7	0.00	1	H1
27	12	40	13	0.00	1	H1
28	18	48	1	0.00	1	H0

Table 15: Industrial CSP instances of Vanderbeck (1999)

inst.	name	m	m'	i	t	n_e	Hi
1	7p18	7	22	13	0.00	0	H2
2	11p4	11	46	12	0.01	1	H1
3	12p19	12	39	13	0.00	1	H1
4	14p12	14	50	15	0.01	1	H1
5	d16p6	16	34	17	0.00	1	H1
6	25p0	25	80	66	0.06	1	H1
7	28p0	28	102	47	0.02	0	H2
8	30p0	26	86	27	0.01	1	H1
9	d33p20	23	53	24	0.06	1	H1
10	d43p21	32	74	33	0.05	1	H1

Table 16: Industrial CSP instances of Degraeve and Schrage (1999)

name	m	m'	i	t	n_e	Hi
DS01	41	90	62	0.06	1	H1
DS02	40	89	71	0.03	0	H2
DS03	26	56	47	0.01	1	H1
DS04	14	29	20	0.00	0	H3
DS05	18	33	31	0.00	0	H2
DS06	71	149	132	0.28	1	H1
DS07	14	41	15	0.01	1	H1
DS08	35	58	47	0.01	1	H1
DS09	35	86	36	0.04	1	H1
DS10	46	98	102	0.06	1	H1
DS11	42	89	43	0.01	1	H1
DS12	53	110	97	0.03	1	H1
DS13	22	47	30	0.00	1	H1
DS14	29	45	40	0.00	0	H0
DS15	43	78	55	0.01	1	H1
DS16	8	24	13	0.00	0	H2
DS17	37	111	36	0.01	0	H2
DS18	16	54	17	0.00	1	H1
DS19	23	67	43	0.02	0	H2
DS20	11	41	18	0.04	0	H2

Table 17: Small-item-size instances with tight KP bounds, *int* = *all*, $\bar{d}$ = *all*

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	26.77	15.30	31	0.00	0.03	113	50	70	0	0	0
20	19.95	53.13	32.52	68	0.01	0.12	113	59	69	0	0	0
30	29.91	79.76	52.21	97	0.04	0.55	130	82	61	0	0	0
40	39.85	105.55	70.60	141	0.10	1.37	126	93	58	0	0	0
50	49.75	132.16	90.50	171	0.20	2.20	132	101	56	0	0	0
75	74.36	197.32	141.59	249	0.61	6.29	154	127	38	0	0	0
100	98.92	263.36	183.75	319	0.88	11.15	153	125	45	0	1	0

Table 18: Medium-item-size instances with tight KP bounds, *int* = *all*, $\bar{d} = 50$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.98	23.09	17.55	29	0.00	0.02	57	48	112	0	1	0
20	19.95	45.58	35.08	58	0.02	0.22	74	57	108	0	0	0
30	29.84	68.47	53.63	91	0.05	0.79	78	77	100	0	1	0
40	39.78	90.65	70.06	116	0.11	3.62	70	64	109	0	0	0
50	49.64	113.69	88.91	154	0.19	4.70	83	71	111	0	3	1
75	74.08	169.10	137.08	216	1.48	53.20	80	71	106	1	0	0
100	98.45	226.07	184.88	293	7.76	410.38	74	72	119	0	2	0

Table 19: CSP instances of Wäscher and Gau (1996) with tight KP bounds, *int* = *all*, $\bar{d}$ = *all*

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	14.13	31	0.00	0.04	452	138	188	0	0	0
20	19.96	50.46	30.56	60	0.01	0.25	467	228	195	0	2	0
30	29.90	75.72	48.17	108	0.03	0.53	479	273	169	0	1	0
40	39.84	100.10	64.79	137	0.08	0.86	499	313	160	0	2	2
50	49.73	125.22	84.25	171	0.17	1.38	522	342	140	0	1	1
all	29.88	75.37	48.38	171	0.06	1.38	2419	1294	852	0	6	3

A.2. Impact of tighter knapsack bounds

The following tables and remarks list only data classes on which the tightening of KP bounds (using (8) instead of (9)) mattered most, giving more details for larger problem sizes.

Concerning Tables 17–18, the good news is that tighter bounds allowed us to solve *all* the small-item-size instances of Degraeve and Peeters (2003), and *all but one* of the medium-item-size instances of Degraeve and Peeters (2003). Unfortunately the running times grew substantially relative to Tabs. 1–2. On the small-item-size instances, for $m \geq 40$ the average running times grew by about 150%; on the medium-item-size instances, the average running times grew by 200%, 217%, 303% and 446% for $m = 40, 50, 75$ and 100 (see Tabs. 20–21 for more details). The iteration numbers were about the same. The increase in running times can be attributed to the knapsack solver (which made more than two million backtrackings

Table 20: Small-item-size instances with tight KP bounds, $\bar{d} = all$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
40	[1, 5000]	39.87	111.88	80.40	141	0.28	1.37	44	42	16	0	0	0
	all	39.85	105.55	70.60	141	0.10	1.37	126	93	58	0	0	0
50	[1, 5000]	49.65	140.88	107.22	171	0.59	2.20	42	41	18	0	0	0
	all	49.75	132.16	90.50	171	0.20	2.20	132	101	56	0	0	0
75	[1, 5000]	74.23	209.83	158.78	236	1.62	6.29	53	53	7	0	0	0
	[1, 7500]	74.67	165.52	189.22	249	0.53	1.79	28	26	16	0	0	0
	[1, 10000]	74.72	142.38	160.50	231	0.19	0.48	15	6	13	0	0	0
	all	74.36	197.32	141.59	249	0.61	6.29	154	127	38	0	0	0
100	[1, 2500]	98.13	359.88	74.83	152	0.16	0.43	59	42	1	0	0	0
	[1, 5000]	98.90	280.47	166.87	295	1.33	11.15	50	47	10	0	0	0
	[1, 7500]	99.18	221.73	268.20	319	1.54	4.55	28	27	23	0	0	0
	[1, 10000]	99.45	191.37	225.10	301	0.51	1.68	16	9	11	0	1	0
	all	98.92	263.36	183.75	319	0.88	11.15	153	125	45	0	1	0

on some subproblems).

Tables 20–21 complement Tables 17–18. Relative to Tabs. 10 and 13, on the small-item-size instances, for $m \geq 40$ the average running times grew mostly from increasing by about 200% on width interval [1, 5000]. On the medium-item-size instances, the average running times increased by 367–531% on width interval [1500, 2500], 157–223% on [500, 5000], and 140–179% on [1000, 5000]; for $m = 100$, they went up by 67–156% on four other intervals. The iteration numbers were about the same.

For the instances of Wäscher and Gau (1996) in Tab. 19, the same 3997 out of 4000 instances were solved, but relative to Tab. 3, for $m = 40$ and 50 the average running times grew by 100% and 143%.

For the instances of Vanderbeck (1999), relative to Tab. 4, the average running times grew by between 67% and 205%; their sum increased by 175%.

Table 21: Medium-item-size instances with tight KP bounds, $\bar{d} = 50$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
30	[1500, 2500]	29.50	88.50	58.50	91	0.29	0.79	12	12	8	0	0	0
	all	29.84	68.47	53.63	91	0.05	0.79	78	77	100	0	1	0
40	[1500, 2500]	39.30	117.90	67.50	101	0.79	3.62	9	9	11	0	0	0
	[500, 5000]	39.85	103.30	82.20	101	0.18	0.31	9	9	11	0	0	0
	all	39.78	90.65	70.06	116	0.11	3.62	70	64	109	0	0	0
50	[1500, 2500]	49.00	147.00	74.95	109	1.11	4.70	4	4	16	0	0	0
	[500, 5000]	49.45	127.80	103.95	128	0.40	0.75	11	11	9	0	0	0
	[1000, 5000]	49.80	121.15	93.15	110	0.20	0.35	8	8	12	0	0	0
	all	49.64	113.69	88.91	154	0.19	4.70	83	71	111	0	3	1
75	[1500, 2500]	71.85	215.55	101.80	159	13.87	53.20	3	3	17	0	0	0
	[500, 5000]	74.10	193.55	164.75	216	1.42	3.15	10	10	9	1	0	0
	[1000, 5000]	74.00	181.05	144.05	175	0.80	1.33	12	12	8	0	0	0
	[1500, 5000]	74.15	167.65	142.70	205	0.43	0.67	9	9	11	0	0	0
	[500, 7500]	74.70	154.65	180.45	216	0.36	0.74	11	11	7	0	0	0
	all	74.08	169.10	137.08	216	1.48	53.20	80	71	106	1	0	0
100	[1500, 2500]	95.55	286.65	128.85	199	84.63	410.38	3	3	17	0	0	0
	[500, 5000]	98.70	263.50	206.20	260	2.61	7.86	17	17	3	0	0	0
	[1000, 5000]	98.65	240.15	197.00	220	1.96	2.75	9	9	11	0	0	0
	[1500, 5000]	98.70	225.25	189.80	263	1.22	1.76	7	7	12	0	1	0
	[500, 7500]	99.15	202.85	255.25	288	0.75	1.21	4	4	12	0	0	0
	[1000, 7500]	99.40	188.45	238.95	293	0.52	0.82	7	9	10	0	0	0
	[1000, 10000]	99.45	163.90	193.25	246	0.31	1.93	2	0	6	0	0	0
	all	98.45	226.07	184.88	293	7.76	410.38	74	72	119	0	2	0

Table 22: Small-item-size instances of Degraeve and Peeters (2003), $\epsilon_r = 0$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	26.77	15.14	31	0.00	0.02	113	49	70	1	0	0
20	19.95	53.13	32.51	69	0.14	22.38	120	64	64	0	0	0
30	29.91	79.76	51.90	91	0.55	127.23	130	85	57	0	0	1
40	39.85	105.55	70.45	134	3.03	219.28	134	98	53	0	0	0
50	49.75	132.16	90.18	181	1.08	168.06	134	102	55	0	0	0
75	74.36	197.32	141.75	256	6.97	576.03	148	121	43	0	1	0
100	98.92	263.36	183.33	311	13.96	1035.61	165	136	33	0	2	0

Table 23: Medium-item-size instances of Degraeve and Peeters (2003), $\epsilon_r = 0$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.98	23.09	17.52	29	0.00	0.01	54	48	112	0	1	0
20	19.95	45.58	35.05	58	0.02	1.80	68	50	114	0	1	0
30	29.84	68.47	53.71	93	0.67	105.02	73	73	105	0	0	0
40	39.78	90.65	69.94	120	3.36	253.23	69	62	111	0	0	0
50	49.64	113.69	88.76	156	1.73	62.13	74	65	118	1	1	1
75	74.08	169.10	137.04	232	30.81	485.60	83	74	104	0	0	1
100	98.45	226.07	184.32	295	67.29	850.75	72	71	118	0	4	0

Table 24: CSP instances of Wäscher and Gau (1996), $\epsilon_r = 0$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	14.27	35	0.00	0.24	449	134	192	0	0	0
20	19.96	50.46	30.73	61	0.13	31.45	485	240	183	0	2	0
30	29.90	75.72	48.18	105	0.43	121.99	503	281	161	0	1	0
40	39.84	100.10	65.05	123	1.60	208.44	503	314	159	0	2	2
50	49.73	125.22	84.75	171	3.13	407.79	526	341	138	0	4	1
all	29.88	75.37	48.60	171	1.06	407.79	2466	1310	833	0	9	3

A.3. Impact of evaluation errors

A.3.1. Comparison with exact bundle

Tables 22–24 summarize our results for exact KP solutions ($\epsilon_r = 0$) relative to Tabs. 1–3 (where $\epsilon_r = 10^{-5}$); similar features were observed on other instances. First, the iteration numbers and the performance of our heuristics did not change significantly. (In other words, the errors occuring in the inexact case were small enough to be accommodated gracefully by our code.) Second, the running times increased quite dramatically. For instance, in Tab. 22 relative to Tab. 1, for $m = 30, 40, 50, 75$ and 100, the average times grew by factors of 27.5, 75.8, 13.5, 29.0 and 34.9, respectively; in Tab. 23 relative to Tab. 2, the factors are 33.5, 112.0, 28.8, 83.3 and 47.1; in Tab. 24 relative to Tab. 3, for $m = 30, 40$ and 50 the factors are 43.0, 40.0 and 44.7. Thus the speedup of inexact bundle ($\epsilon_r = 10^{-5}$) w.r.t. exact bundle

Table 25: Small-item-size instances of Degraeve and Peeters (2003) with tight bounds, $\epsilon_r = 0$

m	$m_{\rm av}$	$m'_{\rm av}$	$i_{\rm av}$	$i_{\rm mx}$	$t_{\rm av}$	$t_{\rm mx}$	n_e	H1	H2	H3	H4	n_g
10	9.99	26.77	15.37	31	0.00	0.03	114	50	69	1	0	0
20	19.95	53.13	32.63	71	0.33	20.99	114	58	70	0	0	0
30	29.91	79.76	52.02	100	0.63	77.87	127	87	55	0	1	0
40	39.85	105.55	70.94	141	2.97	184.59	127	92	59	0	0	0
50	49.75	132.16	91.44	191	1.10	160.34	136	101	56	0	0	0
75	74.36	197.32	142.30	248	25.06	4331.46	148	118	46	1	0	0
100	98.92	263.36	184.39	338	21.27	1694.85	149	119	46	0	6	0

Table 26: Medium-item-size instances of Degraeve and Peeters (2003) with tight bounds, $\epsilon_r = 0$

m	$m_{\rm av}$	$m'_{\rm av}$	$i_{\rm av}$	$i_{\rm mx}$	$t_{\rm av}$	$t_{\rm mx}$	n_e	H1	H2	H3	H4	n_g
10	9.98	23.09	17.55	29	0.00	0.02	57	48	112	0	1	0
20	19.95	45.58	35.08	58	0.03	1.83	74	57	108	0	0	0
30	29.84	68.47	53.63	91	0.67	105.53	78	77	100	0	1	0
40	39.78	90.65	70.05	116	2.98	252.82	69	63	110	0	0	0
50	49.64	113.69	88.92	154	2.37	178.45	83	71	111	0	3	1
75	74.08	169.10	137.08	216	36.34	565.32	80	71	106	1	0	0
100	98.45	226.07	184.73	293	73.35	1043.92	75	73	118	0	2	0

Table 27: CSP instances of Wäscher and Gau (1996) with tight bounds, $\epsilon_r = 0$

m	$m_{\rm av}$	$m'_{\rm av}$	$i_{\rm av}$	$i_{\rm mx}$	$t_{\rm av}$	$t_{\rm mx}$	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	14.15	31	0.00	0.05	450	139	187	0	0	0
20	19.96	50.46	30.73	59	0.35	61.11	474	231	194	0	0	0
30	29.90	75.72	48.41	108	0.23	60.50	484	275	163	0	5	0
40	39.84	100.10	64.91	133	1.75	208.14	513	327	148	0	0	2
50	49.73	125.22	84.78	171	2.31	394.01	536	353	124	0	6	2
all	29.88	75.37	48.60	171	0.93	394.01	2457	1325	816	0	11	4

is (almost uniformly) of order at least 30.

Tabs. 1–3 and 22–24 were obtained for the relaxed bounds of (9). Using the tighter bounds of (8) in exact bundle produced Tabs. 25–27. Here note that, in contrast with inexact bundle which for tighter bounds became at least twice slower on the larger instances (cf. §A.2), the average running times of exact bundle with tighter bounds usually did not increase so much (and sometimes even decreased). Specifically, in Tab. 25 relative to Tab. 22, for $m = 30, 40, 50, 75$ and 100, the average times grew by factors of 1.1, 1.0, 1.0, 3.6 and 1.5, respectively; for Tab. 26 relative to Tab. 23, the factors are 1.0, 0.9, 1.4, 1.2 and 1.1; in Tab. 27 relative to Tab. 24, for $m = 30, 40$ and 50 the factors are 0.5, 1.1 and 0.74.

Hence, the speedups of inexact bundle with relaxed bounds against exact bundle with relaxed or tighter bounds were similar. Indeed, in Tab. 25 relative to Tab. 1, for $m =$

$30, 40, 50, 75$ and 100, the average times grew by factors of 31.5, 74.2, 13.7, 104.4 and 53.2, respectively; for Tab. 26 relative to Tab. 2, the factors are 33.5, 99.3, 39.5, 98.2 and 51.3; in Tab. 27 relative to Tab. 3, for $m = 30, 40$ and 50 the factors are 23.0, 43.1 and 33.0.

Table 28: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d} = all$, $\epsilon_r = 10^{-4}$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.99	26.77	15.14	31	0.00	0.01	113	49	70	1	0	0
20	*all*	19.95	53.13	32.58	70	0.01	0.04	118	62	66	0	0	0
30	*all*	29.91	79.76	52.19	91	0.02	0.23	129	84	58	0	0	1
40	*all*	39.85	105.55	70.91	139	0.04	0.36	133	97	54	0	0	0
50	*all*	49.75	132.16	90.89	211	0.08	0.66	136	104	53	0	0	0
75	*all*	74.36	197.32	143.78	239	0.23	2.00	145	118	47	0	0	0
100	*all*	98.92	263.36	188.89	311	0.38	2.74	163	135	35	0	1	1

Table 29: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$, $\epsilon_r = 10^{-4}$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.98	23.09	17.52	29	0.00	0.01	54	48	112	0	1	0
20	*all*	19.95	45.58	35.04	58	0.01	0.09	68	50	114	0	1	0
30	*all*	29.84	68.47	53.83	93	0.02	0.16	75	75	103	0	0	0
40	*all*	39.78	90.65	70.22	120	0.03	0.57	76	69	104	0	0	0
50	*all*	49.64	113.69	89.36	156	0.06	0.88	79	69	114	1	1	1
75	*all*	74.08	169.10	137.87	232	0.36	8.53	84	75	103	0	0	1
100	*all*	98.45	226.07	186.87	295	1.44	61.50	76	75	114	0	4	0

Table 30: CSP instances of Wäscher and Gau (1996), $int = all$, $\bar{d} = all$, $\epsilon_r = 10^{-4}$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	14.27	35	0.00	0.02	449	134	192	0	0	0
20	19.96	50.46	30.79	61	0.01	0.06	487	242	181	0	2	0
30	29.90	75.72	48.37	105	0.01	0.13	502	280	162	0	1	0
40	39.84	100.10	65.26	136	0.03	0.31	507	318	155	0	2	2
50	49.73	125.22	84.93	171	0.14	60.70	529	344	135	0	4	1
all	29.88	75.37	48.72	171	0.04	60.70	2474	1318	825	0	9	3

A.3.2. Other choices of the relative error tolerance

In parallel with Tabs. 22–24, Tables 28–33 give results for $\epsilon_r = 10^{-4}$ and 10^{-3}. The average iteration numbers and computing times were similar for $\epsilon_r = 10^{-5}$, 10^{-4} and 10^{-3}. However, $\epsilon_r = 10^{-3}$ was too large, causing our heuristics to fail more frequently. On the other hand, $\epsilon_r = 10^{-4}$ did not improve on our standard choice of $\epsilon_r = 10^{-5}$ (giving one more gap in Tab. 28).

Table 31: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d} = all$, $\epsilon_r = 10^{-3}$

m	int	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.99	26.77	15.17	31	0.00	0.01	112	48	71	1	0	0
20	*all*	19.95	53.13	32.22	69	0.00	0.05	111	57	71	0	0	3
30	*all*	29.91	79.76	51.18	91	0.01	0.26	126	80	62	0	0	3
40	*all*	39.85	105.55	69.08	131	0.03	1.05	120	91	60	0	0	5
50	*all*	49.75	132.16	88.54	181	0.05	0.46	122	93	62	0	0	5
75	*all*	74.36	197.32	141.58	242	0.21	5.32	142	119	42	0	3	15
100	*all*	98.92	263.36	196.09	416	0.65	14.45	136	120	48	0	1	22

Table 32: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$, $\epsilon_r = 10^{-3}$

m	int	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.98	23.09	17.55	29	0.00	0.01	52	46	114	0	1	0
20	*all*	19.95	45.58	33.68	52	0.00	0.08	59	42	122	0	1	1
30	*all*	29.84	68.47	51.97	93	0.02	0.33	76	77	101	0	0	1
40	*all*	39.78	90.65	68.33	114	0.03	0.58	74	66	106	0	1	0
50	*all*	49.64	113.69	86.85	178	0.08	2.13	74	69	114	1	1	5
75	*all*	74.08	169.10	133.93	216	0.81	49.70	84	86	92	0	0	13
100	*all*	98.45	226.07	180.35	297	2.79	102.61	68	80	110	0	3	17

Table 33: CSP instances of Wäscher and Gau (1996), $int = all$, $\bar{d} = all$, $\epsilon_r = 10^{-3}$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	14.24	31	0.00	0.02	448	133	193	0	0	0
20	19.96	50.46	30.58	61	0.00	0.08	457	212	208	0	2	3
30	29.90	75.72	47.95	111	0.01	0.26	480	268	170	0	2	6
40	39.84	100.10	64.21	120	0.03	2.70	479	300	167	0	3	17
50	49.73	125.22	82.78	169	0.04	1.27	524	349	126	1	3	16
all	29.88	75.37	47.95	169	0.02	2.70	2388	1262	864	1	10	42

Table 34: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d} = all$, $\epsilon_a = 0.01$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.99	26.77	15.14	31	0.00	0.01	113	49	70	1	0	0
20	*all*	19.95	53.13	32.55	69	0.01	0.05	118	62	66	0	0	0
30	*all*	29.91	79.76	51.95	91	0.02	0.23	129	84	58	0	0	1
40	*all*	39.85	105.55	70.32	134	0.04	0.35	134	98	53	0	0	0
50	*all*	49.75	132.16	90.23	181	0.08	0.66	135	103	54	0	0	0
75	*all*	74.36	197.32	141.99	256	0.24	2.00	146	119	45	0	1	0
100	*all*	98.92	263.36	184.07	311	0.40	2.81	167	138	31	0	2	0

Table 35: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$, $\epsilon_a = 0.01$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.98	23.09	17.52	29	0.00	0.02	54	48	112	0	1	0
20	*all*	19.95	45.58	35.07	58	0.01	0.10	68	50	114	0	1	0
30	*all*	29.84	68.47	53.72	93	0.02	0.16	73	73	105	0	0	0
40	*all*	39.78	90.65	69.97	120	0.03	0.58	74	67	106	0	0	0
50	*all*	49.64	113.69	88.76	156	0.06	0.90	74	65	118	1	1	1
75	*all*	74.08	169.10	137.06	232	0.37	8.60	82	73	105	0	0	1
100	*all*	98.45	226.07	184.43	295	1.44	62.23	71	70	119	0	4	0

Table 36: CSP instances of Wäscher and Gau (1996), *int* = *all*, $\bar{d} = all$, $\epsilon_a = 0.01$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	14.27	35	0.00	0.02	449	134	192	0	0	0
20	19.96	50.46	30.73	61	0.01	0.05	487	242	181	0	2	0
30	29.90	75.72	48.23	105	0.01	0.14	502	280	162	0	1	0
40	39.84	100.10	65.06	123	0.03	0.32	509	320	153	0	2	2
50	49.73	125.22	84.78	171	0.07	0.46	529	344	135	0	4	1
all	29.88	75.37	48.61	171	0.02	0.46	2476	1320	823	0	9	3

A.3.3. Absolute error tolerances

Tables 34–39 give results for $\epsilon_a = 0.01$ and 0.05. For both values of ϵ_a, the average iteration numbers and computing times were close to those in Tabs. 1–3 (where $\epsilon_r = 10^{-5}$). However, $\epsilon_a = 0.05$ was too large, causing our heuristics to fail more frequently. On the other hand, our results for $\epsilon_a = 0.01$ were very close to those for $\epsilon_r = 10^{-5}$.

Table 37: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d} = all$, $\epsilon_a = 0.05$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.99	26.77	15.18	31	0.00	0.01	112	48	71	1	0	0
20	*all*	19.95	53.13	32.50	69	0.00	0.04	116	60	68	0	0	0
30	*all*	29.91	79.76	52.21	91	0.02	0.23	127	82	60	0	0	1
40	*all*	39.85	105.55	71.10	139	0.04	0.35	133	97	54	0	0	0
50	*all*	49.75	132.16	90.96	181	0.08	0.66	135	103	54	0	0	0
75	*all*	74.36	197.32	143.24	256	0.24	2.00	144	117	48	0	0	0
100	*all*	98.92	263.36	184.83	311	0.38	2.81	164	135	35	0	1	0

Table 38: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$, $\epsilon_a = 0.05$

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.98	23.09	17.54	29	0.00	0.01	53	47	113	0	1	0
20	*all*	19.95	45.58	34.66	55	0.01	0.14	64	47	117	0	1	1
30	*all*	29.84	68.47	53.82	93	0.02	0.16	73	73	105	0	0	0
40	*all*	39.78	90.65	70.21	120	0.04	0.59	73	66	107	0	0	0
50	*all*	49.64	113.69	89.41	156	0.06	0.89	75	66	117	1	1	1
75	*all*	74.08	169.10	137.33	232	0.37	8.60	86	77	101	0	0	1
100	*all*	98.45	226.07	185.37	295	1.44	62.23	74	73	116	0	4	0

Table 39: CSP instances of Wäscher and Gau (1996), $int = all$, $\bar{d} = all$, $\epsilon_a = 0.05$

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	14.24	31	0.00	0.02	446	130	195	0	0	1
20	19.96	50.46	30.58	61	0.01	0.14	476	230	191	0	2	2
30	29.90	75.72	48.23	105	0.01	0.60	496	274	167	0	1	1
40	39.84	100.10	65.58	123	0.03	0.78	499	309	163	0	2	3
50	49.73	125.22	84.91	171	0.07	0.46	531	346	132	0	4	2
all	29.88	75.37	48.71	171	0.02	0.78	2448	1289	848	0	9	9

Table 40: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d} = all$, bkmin = 0

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.99	26.77	19.32	51	0.00	0.01	116	52	67	1	0	0
20	*all*	19.95	53.13	43.90	161	0.00	0.04	120	69	59	0	0	0
30	*all*	29.91	79.76	77.51	271	0.01	0.28	126	83	58	0	1	1
40	*all*	39.85	105.55	113.00	430	0.03	0.26	136	103	46	0	2	0
50	*all*	49.75	132.16	156.08	530	0.07	0.56	142	112	44	0	1	0
75	*all*	74.36	197.32	285.02	780	0.22	2.26	150	123	39	0	3	0
100	*all*	98.92	263.36	394.15	1030	0.39	3.65	154	128	40	0	3	2

Table 41: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$, bkmin = 0

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.98	23.09	19.58	50	0.00	0.02	50	42	118	0	1	0
20	*all*	19.95	45.58	44.02	128	0.01	0.08	86	71	93	0	1	0
30	*all*	29.84	68.47	73.53	167	0.01	0.11	96	87	91	0	0	0
40	*all*	39.78	90.65	100.27	261	0.02	0.37	91	87	86	0	0	0
50	*all*	49.64	113.69	135.05	405	0.05	0.56	95	86	94	0	5	1
75	*all*	74.08	169.10	233.36	770	0.36	26.26	100	87	90	0	1	2
100	*all*	98.45	226.07	330.16	990	1.38	104.51	91	91	97	0	5	1

Table 42: CSP instances of Wäscher and Gau (1996), $int = all$, $\bar{d} = all$, bkmin = 0

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	16.75	51	0.00	0.01	439	129	197	0	0	0
20	19.96	50.46	40.19	190	0.00	0.07	487	261	163	0	1	0
30	29.90	75.72	66.79	251	0.01	0.13	498	293	148	0	2	0
40	39.84	100.10	97.90	420	0.02	0.24	534	341	132	0	1	2
50	49.73	125.22	134.75	511	0.05	0.63	542	358	120	0	5	1
all	29.88	75.37	71.28	511	0.02	0.63	2500	1382	760	0	9	3

A.3.4. More inexact null steps

Tables 40–45 give results for bkmin = 0 and 1000 (with $\epsilon_r = 10^{-5}$). Relative to Tabs. 1–3, where bkmin = ∞, for bkmin = 0 the average iteration numbers grew by 59–114% on the largest instances, and four more gaps occured. In contrast, for bkmin = 1000 the average iteration numbers grew by only 5–13% on the largest instances, and three gaps disappeared.

Table 43: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d} = all$, bkmin = 1000

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.99	26.77	15.13	31	0.00	0.01	113	49	70	1	0	0
20	*all*	19.95	53.13	32.77	77	0.00	0.04	123	66	62	0	0	0
30	*all*	29.91	79.76	53.28	131	0.02	0.15	132	88	55	0	0	0
40	*all*	39.85	105.55	73.24	221	0.03	0.26	138	103	48	0	0	0
50	*all*	49.75	132.16	97.07	300	0.07	0.52	143	110	47	0	0	0
75	*all*	74.36	197.32	155.75	376	0.20	1.89	158	125	40	0	0	0
100	*all*	98.92	263.36	202.22	781	0.34	3.62	157	127	42	0	2	0

Table 44: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$, bkmin = 1000

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	*all*	9.98	23.09	17.49	29	0.00	0.01	55	49	111	0	1	0
20	*all*	19.95	45.58	35.26	57	0.01	0.07	77	59	105	0	1	0
30	*all*	29.84	68.47	54.77	101	0.01	0.10	79	79	99	0	0	0
40	*all*	39.78	90.65	71.24	121	0.03	0.38	79	71	102	0	0	0
50	*all*	49.64	113.69	90.22	173	0.05	0.70	86	76	106	1	2	1
75	*all*	74.08	169.10	150.49	533	0.25	6.27	92	84	94	0	0	1
100	*all*	98.45	226.07	208.19	858	0.89	33.81	83	79	110	0	4	0

Table 45: CSP instances of Wäscher and Gau (1996), $int = all$, $\bar{d} = all$, bkmin = 1000

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	H2	H3	H4	n_g
10	9.99	25.37	14.36	35	0.00	0.02	449	134	192	0	0	0
20	19.96	50.46	30.88	65	0.00	0.07	483	240	183	0	2	0
30	29.90	75.72	48.66	111	0.01	0.11	509	291	151	0	1	0
40	39.84	100.10	66.64	171	0.03	0.28	514	323	149	0	3	1
50	49.73	125.22	89.11	306	0.06	0.72	538	351	127	0	5	0
all	29.88	75.37	49.93	306	0.02	0.72	2493	1339	802	0	11	1

Table 46: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d} = all$, H5

m	int	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H5	n_g
10	*all*	9.99	26.77	17.60	39	0.00	0.03	68	119	6
20	*all*	19.95	53.13	36.16	75	0.01	0.03	56	123	14
30	*all*	29.91	79.76	58.37	111	0.02	0.33	49	141	14
40	*all*	39.85	105.55	79.28	143	0.05	0.35	37	150	17
50	*all*	49.75	132.16	102.44	189	0.10	0.64	34	154	20
75	*all*	74.36	197.32	158.43	272	0.27	1.61	31	161	32
100	*all*	98.92	263.36	202.92	311	0.40	2.96	32	169	40

Table 47: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d} = 50$, H5

m	int	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H5	n_g
10	*all*	9.98	23.09	18.76	29	0.00	0.02	15	161	17
20	*all*	19.95	45.58	37.65	58	0.01	0.07	19	165	36
30	*all*	29.84	68.47	56.24	93	0.02	0.17	8	178	63
40	*all*	39.78	90.65	72.87	120	0.04	0.68	9	173	82
50	*all*	49.64	113.69	92.10	156	0.06	0.75	13	185	88
75	*all*	74.08	169.10	140.06	235	0.30	8.56	15	177	111
100	*all*	98.45	226.07	187.75	295	1.24	49.64	5	193	127

A.4. Impact of various heuristics

Tables 46–51 validate the discussion of §5.7 of the paper.

We first consider the case where the heuristics H1 through H4 are replaced by the heuristic named H5, which consists in calling, upon bundle termination, Procedure 1 with Steps 2, 3 and 5 omitted, and Step 4 using FFD; in other words, the relaxed primal solution is rounded down and the residual problem is solved by FFD. The results for H5 (with $\epsilon_r = 10^{-5}$) given in Tables 46–48 show that H5 performs quite poorly relative to Tabs. 1–3 (and that H1 reduces the iteration numbers, and usually the computing times as well). On the other hand, we note that H5 solved 91.5% and 68.8% of problems in Tabs. 46–47, whereas the FFD-based heuristic of Degraeve and Peeters (2003) solved 91.6% and 69.9%; further, H5 solved 92.8% of problems in Tab. 48, whereas the corresponding heuristic RFFD of Wäscher and Gau (1996) solved 92.5%. Thus our bundle results with H5 are very similar to those obtained with other CG solvers.

Our next improvement on H5, named H6, consists in calling Procedure 1 with only Step 2 omitted, and Step 4 using FFD. The results for H6 given in Tables 49–51 show that H6 performs much better than H5, solving 96.4%, 91.9% and 97.2% of problems; thus the rounding procedure of Belov and Scheithauer (2002) may yield significant improvements also for FFD. Finally, we note that H2 and H4 improve on H6 by using SHP or SVC together with Step 2 of Procedure 1. Specifically, H1 and H2 solved 99.8%, 99.4% and 99.7% of problems, and together with H4 they solved 99.94%, 99.88% and 99.92% of problems.

Table 48: CSP instances of Wäscher and Gau (1996), *int* = *all*, $\bar{d}$ = *all*, H5

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	n_g
10	9.99	25.37	15.84	39	0.00	0.02	325	314	26
20	19.96	50.46	34.31	77	0.02	8.36	252	405	48
30	29.90	75.72	54.34	111	0.04	13.00	230	428	56
40	39.84	100.10	73.75	136	0.07	14.42	204	464	69
50	49.73	125.22	95.60	181	0.09	0.55	192	466	90
all	29.88	75.37	54.77	181	0.04	14.42	1203	2077	289

Table 49: Small-item-size instances of Degraeve and Peeters (2003), $\bar{d}$ = *all*, H6

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H5	n_g
10	*all*	9.99	26.77	17.60	39	0.00	0.03	68	120	2
20	*all*	19.95	53.13	36.16	75	0.01	0.04	56	128	3
30	*all*	29.91	79.76	58.37	111	0.02	0.33	49	141	7
40	*all*	39.85	105.55	79.28	143	0.05	0.35	37	151	7
50	*all*	49.75	132.16	102.44	189	0.10	0.64	34	155	7
75	*all*	74.36	197.32	158.43	272	0.27	1.61	31	163	15
100	*all*	98.92	263.36	202.92	311	0.40	2.96	32	171	20

Table 50: Medium-item-size instances of Degraeve and Peeters (2003), $\bar{d}$ = 50, H6

m	*int*	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H5	n_g
10	*all*	9.98	23.09	18.76	29	0.00	0.01	15	161	7
20	*all*	19.95	45.58	37.65	58	0.01	0.06	19	165	4
30	*all*	29.84	68.47	56.24	93	0.02	0.17	8	178	9
40	*all*	39.78	90.65	72.87	120	0.04	0.68	9	173	14
50	*all*	49.64	113.69	92.10	156	0.06	0.75	13	185	25
75	*all*	74.08	169.10	140.06	235	0.30	8.56	15	178	32
100	*all*	98.45	226.07	187.75	295	1.24	49.63	5	193	45

Table 51: CSP instances of Wäscher and Gau (1996), *int* = *all*, $\bar{d}$ = *all*, H6

m	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	H1	n_g
10	9.99	25.37	15.84	39	0.00	0.02	325	325	3
20	19.96	50.46	34.31	77	0.02	8.40	252	416	18
30	29.90	75.72	54.34	111	0.04	12.93	230	435	18
40	39.84	100.10	73.75	136	0.07	14.48	204	471	30
50	49.73	125.22	95.60	181	0.09	0.55	192	476	41
all	29.88	75.37	54.77	181	0.04	14.48	1203	2123	110

Table 52: Small-item-size instances of Degraeve and Peeters (2003), conic bundle

m	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	n_g
10	14.92	32	0.00	0.01	108	0
20	32.66	61	0.01	0.04	110	0
30	53.05	97	0.06	10.63	115	1
40	71.61	140	0.04	0.32	124	0
50	93.20	171	0.09	0.68	139	0
75	145.80	259	0.26	1.89	140	1
100	192.05	338	0.46	4.07	147	0

Table 53: Medium-item-size instances of Degraeve and Peeters (2003), conic bundle

m	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	n_g
10	17.33	27	0.00	0.01	54	0
20	34.92	58	0.01	0.08	63	0
30	53.43	86	0.02	0.14	83	0
40	70.73	123	0.04	0.61	68	0
50	90.10	164	0.07	0.89	69	1
75	139.22	236	0.36	8.28	80	1
100	191.29	300	1.46	59.67	78	0

Table 54: CSP instances of Wäscher and Gau (1996), conic bundle

m	i_{av}	i_{mx}	t_{av}	t_{mx}	n_e	n_g
10	14.24	31	0.00	0.02	425	0
20	31.10	63	0.02	13.13	461	0
30	48.95	110	0.01	0.15	475	0
40	66.34	139	0.04	0.33	513	2
50	86.68	171	0.07	0.58	530	1

A.5. Comparisons with other procedures from the literature

A.5.1. Comparison with Kiwiel and Lemaréchal (2007)

For convenience, Tabs. 52–54 replicate (Kiwiel and Lemaréchal, 2007, Tabs. 1–3). They may be compared with Tabs. 1–3, since they were obtained on the same machine.

A.5.2. Comparison with Degraeve and Peeters (2003)

In Table 55 we compare the average running times of our *bundle relaxation* code BR with the two best procedures HR and LR of Degraeve and Peeters (2003) on the instances used for Tabs. 9, 10, 12 and 13. The times for HR and LR obtained on a Pentium Pro 200 MHz were extracted from (Degraeve and Peeters, 2003, Tabs. 1–4b). Two points should be noted. First, both HR and LR employed an industrial LP solver (much more sophisticated than

Table 55: Comparison of running times with Degraeve and Peeters (2003), *int* = *all*

	Tab. 9			Tab. 10		Tabs. 12–13	
m	HR	LR	BR	LR	BR	LR	BR
30	0.17	0.10	0.02	0.10	0.02	0.29	0.02
40	0.44	0.21	0.05	0.21	0.04	0.71	0.03
50	0.74	0.38	0.08	0.37	0.08	1.45	0.06
75	5.03	0.81	0.22	2.18	0.24	9.57	0.37
100	10.14	2.99	0.42	2.63	0.40	21.08	1.42

our dense QP solver), and LR additionally used subgradient optimization. Second, due to lacking knowledge, let's assume that the machine of Degraeve and Peeters (2003) was ten times slower than ours. Then Table 55 suggests that on the small-item-size instances BR was comparable in speed with HR (about twice slower than LR), while on the medium-item-size instances BR could perform better than LR. Similarly, in view of Tab. 3 and (Degraeve and Peeters, 2003, Tab. 10), on the instances of Wäscher and Gau (1996) BR was as fast as HR (twice slower than LR), whereas Tab. 4 and (Degraeve and Peeters, 2003, Tab. 5a) indicate that on the instances of Vanderbeck (1999) BR was comparable with HR, and sometimes faster than LR. On the industrial instances of Degraeve and Schrage (1999) (cf. Tab. 16 and (Degraeve and Peeters, 2003, Tab. 9)), BR behaved like HR (sometimes better than LR).

A.5.3. Comparison with Briant et al. (2007)

We now compare our running times with those in (Briant et al., 2007, §2.2), where the task was just to produce sufficiently accurate primal and dual solutions $\hat{z}^k$ and $\hat{u}^k$ that satisfy the stopping criteria

$$e\hat{z}^k - \underline{\theta}_k \le \tilde{\epsilon} \quad \text{and} \tag{27a}$$

$$|\pi^k|/\sqrt{m} \le \tilde{\epsilon} \tag{27b}$$

for a given tolerance $\tilde{\epsilon} = 10^{-6}$; thus the duality gap is at most $\tilde{\epsilon}$ and (since $\sum_j p^j \hat{z}^k_j - d \ge \pi^k$) $\hat{z}^k$ satisfies the demand constraints within $\tilde{\epsilon}$ on the average.

However, the first criterion (27a) demands too much in the inexact case: To guarantee that it is eventually met we would need to assume that $\theta^{\infty}_{\hat{u}} - \underline{\theta}_{\infty} < \tilde{\epsilon}$; see below (25). Hence, to achieve a similar implementation context, our code was run with the second test (27b) added to our usual stopping criterion, and without early terminations due to primal heuristics. Table 56 gives our results for the instances of (Briant et al., 2007, §2.2); here "ind_9" comprises the first 9 instances from Tab. 15, "50b100c4" is the final class of Tab. 4 and the remaining classes occur in Tab. 6. The columns "e_{av}" and "e_{mx}" give average and maximum values of *relative dual errors* $|\theta_* - \underline{\theta}_k|/|\theta_*|$ (with θ_* estimated to at least 14 digits in other runs). The columns "a_{av}" and "a_{mx}" give average and maximum values of *absolute errors* a_k, with a_k being the minimum $\tilde{\epsilon}$ satisfying (27); in other words, our code might have terminated earlier if we used (27) as the stopping criterion with $\tilde{\epsilon} \ge a_{\text{mx}}$.

Table 56 was obtained for $\epsilon_r = 0$, i.e., exact KP solutions. The results for $\epsilon_r = 10^{-5}$ are given in Tab. 57, and for $\epsilon_r = 10^{-4}$ and 10^{-3} in Tabs. 58–59. The accuracy obtained was

Table 56: Industrial and random CSP instances of Briant et al. (2007), $\epsilon_r = 0$

name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	e_{av}	e_{mx}	a_{av}	a_{mx}
ind_9	18.00	56.89	31.11	73	0.02	0.10	4.4E-16	1.0E-15	4.5E-12	3.1E-11
50b100c4	49.70	129.25	109.80	145	0.16	0.44	2.3E-16	7.8E-16	3.0E-12	1.4E-11
u120	63.20	88.75	98.75	124	0.02	0.03	5.2E-15	4.8E-14	6.3E-13	4.2E-12
u250	77.25	129.00	107.85	139	0.02	0.04	3.8E-15	3.0E-14	2.9E-12	1.5E-11
t120	86.15	110.75	76.05	93	0.02	0.03	0.0E+00	0.0E+00	1.3E-12	6.5E-12
t249	140.10	199.15	133.20	148	0.05	0.06	0.0E+00	0.0E+00	8.4E-08	1.7E-06

Table 57: Industrial and random CSP instances of Briant et al. (2007), $\epsilon_r = 10^{-5}$

name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	e_{av}	e_{mx}	a_{av}	a_{mx}
ind_9	18.00	56.89	31.67	73	0.03	0.10	4.4E-16	1.0E-15	9.3E-11	7.9E-10
50b100c4	49.70	129.25	109.80	145	0.16	0.45	2.3E-16	7.8E-16	3.0E-12	1.4E-11
u120	63.20	88.75	99.55	124	0.02	0.03	5.2E-15	4.8E-14	7.2E-11	9.0E-10
u250	77.25	129.00	108.60	139	0.02	0.04	2.3E-15	3.0E-14	1.3E-10	9.4E-10
t120	86.15	110.75	78.00	95	0.02	0.02	0.0E+00	0.0E+00	4.2E-11	1.8E-10
t249	140.10	199.15	134.90	153	0.05	0.06	0.0E+00	0.0E+00	1.3E-11	2.2E-10

Table 58: Industrial and random CSP instances of Briant et al. (2007), $\epsilon_r = 10^{-4}$

name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	e_{av}	e_{mx}	a_{av}	a_{mx}
ind_9	18.00	56.89	32.22	70	0.02	0.10	4.4E-16	1.0E-15	2.2E-05	2.0E-04
50b100c4	49.70	129.25	109.80	145	0.16	0.45	2.3E-16	7.8E-16	3.0E-12	1.4E-11
u120	63.20	88.75	98.45	124	0.02	0.03	5.2E-15	4.8E-14	1.2E-10	1.2E-09
u250	77.25	129.00	110.50	144	0.02	0.04	4.3E-15	3.2E-14	1.5E-05	1.7E-04
t120	86.15	110.75	76.85	98	0.02	0.02	0.0E+00	0.0E+00	2.7E-06	3.0E-05
t249	140.10	199.15	135.00	154	0.05	0.06	0.0E+00	0.0E+00	9.0E-06	5.5E-05

Table 59: Industrial and random CSP instances of Briant et al. (2007), $\epsilon_r = 10^{-3}$

name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	e_{av}	e_{mx}	a_{av}	a_{mx}
ind_9	18.00	56.89	29.00	54	0.02	0.10	3.7E-07	3.3E-06	1.0E-02	4.3E-02
50b100c4	49.70	129.25	98.10	112	0.08	0.15	2.5E-05	1.4E-04	6.8E-02	2.8E-01
u120	63.20	88.75	103.85	127	0.02	0.03	1.0E-14	1.8E-13	3.7E-04	4.4E-03
u250	77.25	129.00	118.70	143	0.03	0.05	4.0E-06	4.8E-05	4.8E-03	1.6E-02
t120	86.15	110.75	87.85	98	0.02	0.03	0.0E+00	0.0E+00	1.4E-03	5.1E-03
t249	140.10	199.15	149.05	161	0.06	0.07	0.0E+00	0.0E+00	2.0E-03	6.0E-03

Table 60: Comparison with Briant et al. (2007), $\epsilon_r = 10^{-5}$

name	t_{av}				speedup
	KBASIC	KRICH	BUNDLE	BR	
ind_9	0.52	20.91	42.68	0.03	17.3
50b100c4	4.51	3.72	27.17	0.16	23.2
u120	1.79	1.15	1.36	0.02	57.5
u250	2.90	2.03	1.60	0.02	80.0
t120	7.83	4.14	2.84	0.02	142.0
t249	61.36	16.49	9.09	0.05	181.8

Table 61: CSP instances of Briant et al. (2007) with tight KP bounds, $\epsilon_r = 0$

name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	e_{av}	e_{mx}	a_{av}	a_{mx}
ind_9	18.00	56.89	31.44	70	0.09	0.58	3.3E-16	9.3E-16	3.6E-12	2.3E-11
50b100c4	49.70	129.25	109.05	145	0.47	1.55	3.4E-16	1.4E-15	2.4E-12	1.4E-11
u120	63.20	88.75	97.35	123	0.02	0.04	6.2E-15	8.9E-14	9.0E-13	9.8E-12
u250	77.25	129.00	107.60	137	0.03	0.04	4.6E-15	6.4E-14	3.3E-12	9.6E-12
t120	86.15	110.75	78.15	93	0.02	0.02	0.0E+00	0.0E+00	1.2E-12	6.5E-12
t249	140.10	199.15	132.65	145	0.05	0.06	0.0E+00	0.0E+00	2.2E-12	1.1E-11

Table 62: CSP instances of Briant et al. (2007) with tight KP bounds, $\epsilon_r = 10^{-5}$

name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	e_{av}	e_{mx}	a_{av}	a_{mx}
ind_9	18.00	56.89	31.11	70	0.09	0.57	3.3E-16	9.3E-16	4.9E-12	3.4E-11
50b100c4	49.70	129.25	109.05	145	0.47	1.55	3.4E-16	1.4E-15	2.4E-12	1.4E-11
u120	63.20	88.75	97.60	123	0.02	0.04	6.2E-15	8.9E-14	1.1E-11	2.0E-10
u250	77.25	129.00	109.05	141	0.03	0.04	1.5E-15	7.6E-15	4.4E-10	3.5E-09
t120	86.15	110.75	81.05	95	0.02	0.03	0.0E+00	0.0E+00	1.0E-10	7.6E-10
t249	140.10	199.15	134.35	148	0.05	0.06	0.0E+00	0.0E+00	4.8E-11	4.7E-10

quite poor for $\epsilon_r = 10^{-3}$, a bit too weak for $\epsilon_r = 10^{-4}$, but very good for $\epsilon_r = 10^{-5}$ (the results for smaller ϵ_r were similar).

In view of the excellent accuracy in Tab. 57, in Tab. 60 we compare our *bundle relaxation* code BR using $\epsilon_r = 10^{-5}$ with the three codes KBASIC, KRICH and BUNDLE of (Briant et al., 2007, Tabs. 1, 2 and 5), which implement two CG variants and an exact bundle variant respectively. We show the average running times "t_{av}" and the speedup of BR with respect to the fastest code of Briant et al. (2007), where the machine used was about twice slower than ours.

Note that on this fairly small set of 109 instances, there is little difference between exact bundle (Tab. 56) and standard inexact bundle (Tab. 57). Yet even this small set can illustrate the advantages of KP bound relaxation, another ingredient of our approach. Namely, Tabs. 56–59 were obtained for the relaxed bounds of (9). For the tighter bounds of (8), Tabs. 61–64 exhibit significant slow downs on the first two classes ind_9 and 50b100c4.

Table 63: CSP instances of Briant et al. (2007) with tight KP bounds, $\epsilon_r = 10^{-4}$

name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	e_{av}	e_{mx}	a_{av}	a_{mx}
ind_9	18.00	56.89	31.67	66	0.09	0.57	3.3E-16	9.3E-16	1.2E-05	1.1E-04
50b100c4	49.70	129.25	109.05	145	0.47	1.55	3.4E-16	1.4E-15	2.4E-12	1.4E-11
u120	63.20	88.75	98.35	123	0.02	0.04	6.3E-15	8.9E-14	1.4E-10	1.4E-09
u250	77.25	129.00	109.65	133	0.03	0.04	5.0E-15	5.7E-14	1.1E-05	1.3E-04
t120	86.15	110.75	77.60	94	0.02	0.02	0.0E+00	0.0E+00	2.7E-06	2.0E-05
t249	140.10	199.15	136.50	159	0.05	0.08	0.0E+00	0.0E+00	9.8E-06	8.8E-05

Table 64: CSP instances of Briant et al. (2007) with tight KP bounds, $\epsilon_r = 10^{-3}$

name	m_{av}	m'_{av}	i_{av}	i_{mx}	t_{av}	t_{mx}	e_{av}	e_{mx}	a_{av}	a_{mx}
ind_9	18.00	56.89	28.44	47	0.07	0.58	3.7E-07	3.3E-06	1.1E-02	4.7E-02
50b100c4	49.70	129.25	99.10	120	0.23	0.56	3.2E-05	1.5E-04	8.3E-02	3.4E-01
u120	63.20	88.75	105.40	142	0.02	0.04	3.9E-15	4.3E-14	3.5E-04	3.4E-03
u250	77.25	129.00	117.75	142	0.03	0.04	1.6E-06	1.7E-05	6.2E-03	2.2E-02
t120	86.15	110.75	87.40	96	0.02	0.02	0.0E+00	0.0E+00	1.1E-03	2.8E-03
t249	140.10	199.15	148.45	172	0.06	0.08	0.0E+00	0.0E+00	3.2E-03	1.7E-02