APLICATIONES MATHEMATICAE
23, 2 (1995) 151–167

A branch&bound algorithm for solving one-dimensional cutting stock problems exactly

Guntram Scheithauer and Johannes Terno

Dresden University of Technology

Abstract

Many numerical computations reported in the literature show an only small difference between the optimal value of the one-dimensional cutting stock problem (1CSP) and that of its corresponding linear programming relaxation. Moreover, theoretical investigations have proven that this difference is smaller than 2 for a wide range of subproblems of the general 1CSP.

In this paper we give a branch&bound algorithm to compute optimal solutions for instances of the 1CSP. Numerical results are presented of about 900 randomly generated instances with up to 100 small pieces and all of them are solved to optimality.

Key words: Integer Optimization, Cutting Stock Problem, branch&bound, Rounding

Classification: 90 C 10, 90 C 05

1 Introduction

The one-dimensional cutting stock problem (1CSP) is the following:

> One-dimensional material objects of a given length L are divided into smaller pieces of desired lengths $l_1, \ldots, l_m$ in order to fulfill the order demands $b_1, \ldots, b_m$. The goal is to minimize the total amount of stock material or, equivalently, to minimize the total waste.

It is well known [6] that the 1CSP can be modelled as a linear integer optimization problem as follows. Any feasible cutting pattern can be represented by an m-dimensional nonnegative integer vector $a^j = (a_{1j}, \ldots, a_{mj})^T$ fulfilling $\sum_{i=1}^m l_i a_{ij} \leq L$. Defining integer variables x_j to give the number of stock material to be cut according to pattern a^j one has

$$z = \sum_{j=1}^n x_j \ \rightarrow \ \min$$

$$\text{s.t.} \qquad \sum_{j=1}^n a_{ij} x_j \ \geq \ b_i, \quad i = 1, \ldots, m,$$

$$x_j \geq 0, \ \text{integer}, \ j = 1, \ldots, n,$$

where n denotes the number of cutting patterns. Without loss of generality, all Input-data can be assumed to be integer, and in order to ensure solvability, we suppose $\max_{i=1,\ldots,m} l_i \leq L$. Furthermore we assume $b_i \geq 1$ for all i and $l_1 > l_2 > \ldots > l_m$.

This model can be written in the short form

$$z = e^T x \ \rightarrow \ \min \quad \text{s.t.} \qquad Ax \ \geq \ b, \quad x \in Z_+^n \tag{P}$$

with $e = (1, \ldots, 1)^T$ and where the coefficient matrix A contains either all feasible cutting patterns or, in an equivalent model, all maximal cutting patterns a^j as columns. A pattern $a \in Z_+^m$ is called maximal if $0 \leq L - l^T a < \min_{i=1,\ldots,m} l_i$. In general we consider the case where only the maximal cutting patterns are contained in the matrix. In case an instance (m, l, L, b) of the 1CSP is given then the instance (m, n, A, c, b) for model (P) is uniquely determined but the reverse is not true (cf. in [15]).

Because of the in general exponential number n of variables and the integrality condition the 1CSP is at least NP-hard. For that reason, a frequently used solution strategy to obtain nearly optimal integer solutions consists in solving the linear programming (LP) relaxation

$$z = e^T x \ \rightarrow \ \min \quad \text{s.t.} \qquad Ax \ \geq \ b, \quad x \in R_+^n \tag{Q}$$

of (P) using the revised simplex method with column generation and an appropriate rounding (cf. [6], [7], [16]).

On the other hand, many numerical computations (cf. [13], [17]) show an only small difference between the optimal value of the 1CSP (denoted by $z^*(E)$) and that of its corresponding LP relaxation (denoted by $z_c(E)$). Up to now no instance E with $z^*(E) > \lceil z_c(E) \rceil + 1$ was found [5], [10], [13], [14].

Moreover, in [1] and [14] investigations are reported with respect to the gap between the optimal value $z^*(E)$ and $z_c(E)$ for instances E of the 1CSP. There the so-called integer round-up property (IRUP) and the modified integer round-up property (MIRUP) were proven for several subproblems of the 1CSP . And there is a founded hope that the MIRUP holds true for the general 1CSP.

The aim and the organization of this paper are as follows. Based on these investigations for the 1CSP a solution strategy is proposed which is directly oriented on the MIRUP conjecture. In order to compute an optimal (integer) solution for an instance of the 1CSP the solution strategy (Section 3) uses at first a reduction (Section 2) of the instance. For that purpose an optimal or nearly optimal solution of the continuous relaxation problem is to compute. After that a very effective greedy strategy secures in the most cases the determination of an optimal solution. In the other cases a branch&bound algorithm is applied to the reduced instance (Section 4, 6). In order to improve the performance of the solution process a termination criterion for computing LP bounds using the simplex method is discussed in Section 5. Numerical experiments with randomly generated instances (having up to 100 small pieces) show the efficiency of the proposed algorithm (Section 7).

2 Problem reduction

Baum and Trotter [1] define that an integer minimization problem of type (P) is said to have the integer round-up property (IRUP) if for any instance E its optimal value $z^*(E)$ is given by the smallest integer greater than or equal to the optimal value of its LP relaxation, i.e.[1] $z^*(E) = \lceil z_c(E) \rceil$. It is well known [10], [5], [12] that the 1CSP does not belong to the class of problems having the IRUP.

In [12] the modified integer round-up property (MIRUP) is defined. An integer minimization problem is said to have the MIRUP if for any instance E the optimal value is bounded from above by the LP lower bound rounded up plus 1, i.e. $z^*(E) \leq \lceil z_c(E) \rceil + 1$. Furthermore the conjecture whether the general 1CSP has the MIRUP is numerically investigated in [13].

Let $\mathcal{M}$ denote the set of all instances of the 1CSP having the MIRUP, and let $\mathcal{M}^*$ be the set of those instances having the IRUP.

In order to investigate the 1CSP with respect to the IRUP or the MIRUP some reductions of the right hand side can be made similar to [14]

Let $E = (m, l, L, b)$ be an instance of the 1CSP with coefficient matrix A and let x^c denote an optimal solution of the LP relaxation (Q) of P. Rounding down yields an integer vector $\underline{x}$ with[2] $\underline{x}_j = \lfloor x_j^c \rfloor$ and a real vector of fractional parts $\{x_j\} = x_j^c - \underline{x}_j$, $j = 1, \ldots, n$. If $\underline{x} \neq x^c$ then a residual instance can be defined with the right hand side $\overline{b} := b - A\underline{x}$. Hence, the residual instance $\overline{E} := (m, l, L, \overline{b})$ is also an instance of the 1CSP.

Lemma 1 *Let E be an instance of the 1CSP and $\overline{E}$ a corresponding residual instance. Then it holds:*

[1] $\lceil x \rceil$ denotes the smallest integer not smaller than x.
[2] $\lfloor x \rfloor$ denotes the largest integer not larger than x.

a)	$\overline{E} \in \mathcal{M}^*$	$\Rightarrow$	$E \in \mathcal{M}^*$,
b)	$\overline{E} \in \mathcal{M}$	$\Rightarrow$	$E \in \mathcal{M}$.

Proof: It holds:

a) $z^*(E) \leq e^T \underline{x} + z^*(\overline{E}) \leq e^T \underline{x} + \lceil z_c(\overline{E}) \rceil = \lceil e^T \underline{x} + z_c(\overline{E}) \rceil = \lceil z_c(E) \rceil$.

b) $z^*(E) \leq e^T \underline{x} + z^*(\overline{E}) \leq e^T \underline{x} + \lceil z_c(\overline{E}) \rceil + 1 = \lceil e^T \underline{x} + z_c(\overline{E}) \rceil + 1 = \lceil z_c(E) \rceil + 1$.

A generalization of Lemma 1 implies the following problem reduction:

Lemma 2 *Let $E = (m, l, L, b)$ be an instance of the 1CSP with coefficient matrix A and let $x^s \in R^n_+$ be such that $Ax^s \geq b$ and $e^T x^s \leq \lceil z_c(E) \rceil$. If the residual problem $\overline{E} := (m, l, L, b - A\lfloor x^s \rfloor)$ has a solution $x^r \in Z^n_+$ with*

a) $e^T x^r \leq \lceil e^T x^s \rceil$ *then there exists a solution $x^* \in Z^n_+$ of E with* $e^T x^* \leq \lceil z_c(E) \rceil$, *i.e.* $E \in \mathcal{M}^*$,

b) $e^T x^r \leq \lceil e^T x^s \rceil + 1$ *then there exists a solution $x^* \in Z^n_+$ of E with* $e^T x^* \leq \lceil z_c(E) \rceil + 1$, *i.e.* $E \in \mathcal{M}$.

Proof: We set $x^* := \lfloor x^s \rfloor + x^r$. Then $x^* \in Z^n_+$ and

$Ax^* = A\lfloor x^s \rfloor + Ax^r \geq A\lfloor x^s \rfloor + b - A\lfloor x^s \rfloor = b$. Furthermore,

a) $e^T x^* = e^T \lfloor x^s \rfloor + e^T x^r \leq e^T \lfloor x^s \rfloor + \lceil e^T x^s \rceil = \lceil e^T x^s \rceil \leq \lceil z_c(E) \rceil$,

b) $e^T x^* = e^T \lfloor x^s \rfloor + e^T x^r \leq e^T \lfloor x^s \rfloor + \lceil e^T x^s \rceil + 1 = \lceil e^T x^s \rceil + 1 \leq \lceil z_c(E) \rceil + 1$.

Hence, if an optimal solution of the residual instance is found then an optimal solution of the initial instance can be constructed or at least an integer solution is obtained proving the MIRUP. Moreover Lemma 2 shows that only a nearly optimal solution of the continuous relaxation of the initial instance is necessary to construct a suitable residual instance.

In order to prove the validity of the IRUP or the MIRUP we have the following lemma for special sets of instances.

Lemma 3 *Let $E = (m, l, L, b)$ be a residual instance.*

a) If $l^T b \leq 1.5L$ *or* *if* $2L < l^T b \leq 2.5L$ *then* $E \in \mathcal{M}^*$.

b) If $l^T b \leq 3L$ *then* $E \in \mathcal{M}$.

c) If $z_c(E) > m - 1$ *then* $E \in \mathcal{M}^*$.

d) If $z_c(E) > m - 2$ *then* $E \in \mathcal{M}$.

Most of these statements are proven in [14].

3 Solution concept

Let be given an instance $E = (m, l, L, b)$ of the 1CSP. Based on the possibilities of reduction at first the corresponding LP relaxation (Q) is solved until either an optimal solution x is found or a feasible solution x with $e^T x \leq \lceil z_c(E) \rceil$ is obtained. After that a residual instance $\overline{E} = (\overline{m}, \overline{l}, L, \overline{b})$ is defined where $\underline{x} := \lfloor x \rfloor$, $\overline{m} := \sum_{i=1}^{m} sign(\max\{0, [A\underline{x}]_i - b_i\})$ and $\overline{l}$ and $\overline{b}$ consist of the corresponding piece lengths and reduced order quantities, respectively.

In case x is an optimal solution of (Q) it holds $z_c(\overline{E}) = e^T x - e^T \underline{x}$.

Applying two heuristics a feasible integer solution x^h of $\overline{E}$ with value $z^h = e^T x^h$ is constructed. An optimal solution of E is found if $z^h \leq \lceil z_c(\overline{E}) \rceil$ holds. Otherwise, the residual problem $\overline{E}$ has to be solved exactly. Because of the NP-hardness of the 1CSP a branch&bound algorithm is used. If $z^*(\overline{E}) = \lceil z_c(\overline{E}) \rceil$ then an optimal solution of E is known.

In the case $z^*(\overline{E}) \geq \lceil z_c(\overline{E}) \rceil + 1$ there is to decide on hands of sufficient conditions that $z^*(E) \geq \lceil z_c(E) \rceil + 1$ follows, or the branch&bound algorithm has to be applied to a somewhat extended problem until a solution x^* of E is found with $z^*(E) = \lceil z_c(E) \rceil$.

That there exist instances E of the 1CSP with

$$z^*(E) \ < \ \underline{z} + z^*(\overline{E})$$

can be illustrated on hands of the following instance. Let $L = 396,\ l_1 = 132,\ l_2 = 99,\ l_3 = 44,\ l_4 = 36$ with $b_1 = 2,\ b_2 = 3,\ b_3 = 9$ and $b_4 = 6$. The continuous solution is

$$\frac{2}{3}\begin{pmatrix} 3 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \frac{3}{4}\begin{pmatrix} 0 \\ 4 \\ 0 \\ 0 \end{pmatrix} + 1\begin{pmatrix} 0 \\ 0 \\ 9 \\ 0 \end{pmatrix} + \frac{6}{11}\begin{pmatrix} 0 \\ 0 \\ 0 \\ 11 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 9 \\ 6 \end{pmatrix}$$

with $z_c(E) = \dfrac{391}{132}$ since the applied patterns contain no waste. It is $\underline{z} = 1$. The residual problem $\overline{E}$ with $\overline{b} = (2, 3, 0, 6)^T$ has the optimal value $z^*(\overline{E}) = 3$ but $z^*(E) = 3$.

4 Lower bounds and heuristics

In order to solve one-dimensional cutting stock problems exactly a branch & bound algorithm is proposed which is only applied to residual instances. Three bounds are used. Let $E^r = (m_r, l^r, L, V, b^r)$ denote a subproblem generated in the branch&bound algorithm where the level r gives the number of fixed cutting patterns having occurrence greater than 0 and where V denotes a set of forbidden cutting patterns (occurrence 0).

The first lower bound is the natural material bound

$$bound_1(E^r) := \lceil \frac{(l^r)^T b^r}{L} \rceil.$$

The second lower bound used is derived from an adapted LP relaxation and is obtained by solving the problem

$$\overline{z}_c(E^r) \ := \ \min \{ \ e^T x : \quad Ax \geq b^r, \ x_j \geq 0, \ a^j \not\leq b^r \Rightarrow x_j = 0, \quad a^j \in V \Rightarrow x_j = 0, \ (j = 1, \ldots, n)\}. \qquad (\overline{Q}^r)$$

In order to lower the computational effort the number of simplex steps (of solved knapsack problems) is limited by $6m$ if $r = 0$, and by $2m$ if $r > 0$. Let $\overline{z}(E^r)$ denote the value computed with $6m$ or $2m$ simplex steps, respectively. The exactness of $\overline{z}(E^r)$ to be a lower bound of $z^*(E^r)$ can be verified using the termination criterion which is defined in the next section. In some cases the exactness of $\overline{z}(E^r)$ cannot be proven. Therefore the second bound is defined as follows:

$$bound_2(E^r) := \left\{ \begin{array}{ccl} \lceil \overline{z}(E^r) \rceil & : & \overline{z}(E^r) \text{ is valid bound} \\ bound_2(E^{r-1}) & : & \text{otherwise} \end{array} \right\}.$$

The third lower bound is obtained by using the quotient from the actual order demands and a cutting pattern which is the solution of a knapsack problem (K) with weights $k_i := \lfloor L/l_i \rfloor$. Let define the problem (K) and the amount $\gamma(b)$:

$$(K) \qquad \mu(b) := \max \{ \sum_{i=1}^{m} \frac{a_i}{k_i} \ : \ l^T a \leq L, \ a \leq b, \ a \notin V, \ a \geq 0, \ integer\}$$

$$\gamma(b) := \sum_{i=1}^{m} \frac{b_i}{k_i}$$

Then

$$bound_3(E^r) := \lceil \frac{\gamma(b^r)}{\mu(b^r)} \rceil$$

is a lower bound of $z^*(E^r)$ (cf. [14]).

Now we describe two heuristics to get a feasible cutting pattern $a \in Z_+^m$ for an instance $E = (m, l, L, b)$ of the 1CSP with the right hand side b.

In the first heuristic the cutting pattern is constructed by using a direct greedy method.

Heuristic 1 (L, b, a);

- $\Delta L := L$, $\Delta b := b$;
- for $i := 1$ to m do
  $a_i := \min\{\Delta b_i, \lfloor \frac{\Delta L}{l_i} \rfloor\}$; $\Delta b_i := \Delta b_i - a_i$; $\Delta L := \Delta L - a_i \cdot l_i$;

In the second heuristic the cutting pattern is constructed by using a modified greedy method. Let $\zeta := z_c(E) - e^T \lfloor x^c(E) \rfloor$. That means, ζ is the sum of all fractional parts of the optimal solution x^c of the corresponding linear relaxation problem. Using the "weight" $1/\zeta$ a more equalized cutting pattern is constructed in comparison to Heuristic 1.

Heuristic 2 (L, b, a);

- $\Delta L := L$; $\Delta b := b$, $\zeta := \max\{1, \zeta\}$;
- for $i := 1$ to m do
  $a_i := \min\{\lceil \frac{\Delta b_i}{\zeta} \rceil, \lfloor \frac{\Delta L}{l_i} \rfloor\}$; $\Delta b_i := \Delta b_i - a_i$ $\Delta L := \Delta L - a_i \cdot l_i$;
- if $L - l^T a \geq l_m$ then Heuristic1 $(\Delta L, \Delta b, \Delta a)$, $a := a + \Delta a$;

In order to get a feasible solution for the instance E the first or second heuristic are repeatedly applied until $\Delta b = 0$.

5 Termination criterion

Using the LP relaxation to get lower bounds for problem (P) one has to overcome the difficulties which arise from the unknownness of the coefficient matrix A. Applying the primal (revised) simplex method for (Q) a valid lower bound is obtained not before (Q) is solved exactly or a feasible solution $x \in R^n_+$ with value $z = e^T x$ is found such that $z \leq \lceil z_c(b) \rceil$ where $z_c(b)$ denotes the optimal value of (Q) for right hand side b..

Because A is not available explicitly it is impossible to compute lower bounds by using the dual problem of (Q).

Since for instances of medium size the number of simplex steps needed varies in a wide range and problems of numerical stability may occur (the objective function value decreases very slowly within a block of simplex steps) it is sometimes not advantageous to continue the column generation process until the optimality criterion of the simplex method is fulfilled. Since it is sufficient to have a feasible solution $x \in R^n_+$ with value $z = e^T x$ and

$$z \leq \lceil z_c(b) \rceil \tag{1}$$

we need a criterion to decide whether (1) is fulfilled or not. Such a criterion can be obtained using FARKAS'Lemma [11]. We consider the problem whether there exists a vector $x \in R^n_+$ with

$$-Ax \leq -b, \quad e^T x = z_0 \tag{2}$$

where $z_0 := \lfloor z \rfloor$. Supposing $z_0 < z_c(b)$ then there is no solution of problem (2). Hence, FARKAS'Lemma yields

Lemma 4 *The system of inequalities*

$$\begin{aligned} -A^T u + u_0 e &\geq 0, \\ b^T u - z_0 u_0 &> 0, \\ u &\geq 0 \end{aligned} \tag{3}$$

has a feasible solution (u_0, u) *if and only if* $z_0 < z_c(b)$.

Hence, if the feasible solution x of (Q) fulfilles (1) and $z_c(b) \notin Z$ then (3) is solvable for $z_0 := \lfloor z \rfloor$. On the other hand, if x does not fulfill (1) then (3) has no solution for $z_0 := \lfloor z \rfloor$.

Let us assume that (1) is fulfilled. Then because of (3)

$$b^T u - z_0 u_0 \geq \varepsilon > 0.$$

Substituting u_i by εu_i it follows

$$b^T u - z_0 u_0 \geq 1.$$

Therefore the solvability of (3) is equivalent to the solvability of the minimization problem

$$\begin{aligned} & w = u_0 \to \min \\ \text{s.t.} \quad & u_0 e - A^T u \geq 0, \ -z_0 u_0 + b^T u \geq 1, \\ & u \geq 0, \end{aligned} \tag{4}$$

where the matrix A^T consists of all maximal cutting patterns as rows. Hence "row generation" is required, i.e. we have to choose m linear independent rows (cutting patterns) a^j which form a matrix $\overline{A} = (a^1, \ldots, a^m)$. Solving problem (4) with coefficient matrix $\overline{A}^T$ instead of A^T leads to the solution (u_0, u). If

$$\max\{u^T a \ : \ l^T a \leq L, \ a \in Z_+^m\} \ \leq \ u_0$$

then a solution of (4) is found. Otherwise a new row can be inserted in the basis matrix.

Now we discuss the application of Lemma 4 in the process of solving (Q). In a certain step we want to proof the validity of the condition (1) in order to stop the column generation. Therefore we try to solve the system of inequalities (3) or the problem (4), respectively. The process of solving (Q) is to continue if after a given number of generation steps no decision whether (3) is solvable or not is found. Especially in the case $z_c(b) \in Z$ only the optimality criterion of the simplex method works. On the other hand, in the case when (1) could not be proven then most of the generated rows are useful for the solution process of (Q). By this the computational effort for solving (4) is dominated by its usefulness with respect to the total solution process.

6 The branch&bound algorithm

In this section a branch & bound algorithm is described to solve a residual instance $E = (m, l, L, b)$ of the one-dimensional cutting stock problem.

In the algorithm a parameter β with $\beta \in \{1, 2, \ldots\}$ controls the computation of LP bounds for subproblems. Furthermore two branching rules are used which also depend on β and differ in choosing a cutting pattern to be fixed. Both are bisection methods.

Within the algorithm the "level" r gives the number of fixed cutting patterns with occurrence 1, and s_r is the number of all fixed cutting patterns up to level

r including those having occurrence 0. The latter are called forbidden cutting patterns.

In order to control the computation of LP bounds ($bound_2$) the parameter β is used as follows. Only if $s_r \bmod \beta = 0$ then the LP bound is computed for the current subproblem. Hence, if $\beta = 1$ then for each subproblem a LP bound is to compute.

In the case $s_r \bmod \beta = 0$ the next cutting pattern a^j to be fixed is chosen from the current LP solution x^c accordingly to a maximal x_j^c, i.e. $x_j^c = \max\{x_k^c : k = 1, 2, \ldots, n\}$.

If $\beta > 1$ then the two branching rules, which define variants (a) and (b), are as follows.

> Variant (a): "LP-cutting-pattern-strategy"
> If $s_r \bmod \beta \neq 0$ then the next cutting pattern a^* to be fixed is chosen from the last LP solution x^c computed accordingly to a maximal x^c-value and not considered before. (If such a pattern does not exist a new LP bound has to be computed.)
>
> Variant (b): "High-density-cutting-pattern-strategy"
> If $s_r \bmod \beta \neq 0$ then the next cutting pattern a^* to be fixed is the solution of the following knapsack problem. Let $E^r = (m_r, l^r, L, b^r)$ denote the current subproblem and let $k_i := \lfloor L/l_i^r \rfloor, \ i = 1, \ldots, m_r$.

$$\max \left\{ \sum_{i=1}^{m_r} \frac{a_i}{k_i} \ : \ \sum_{i=1}^{m_r} l_i^r a_i \leq L, \ 0 \leq a_i \leq b_i^r, \ integer \right\}.$$

Let define the following sets and variables used in the branch&bound algorithm to solve the residual instance $E = (m, l, L, b)$:

r	:	The level of a node within the branching tree, number of fixed cutting patterns with occurrence 1.
s_r	:	Number of all fixed cutting patterns.
C	:	A set which contains all fixed cutting patterns with occurences 1.
V	:	A set which contains all forbidden cutting patterns.
V^r	:	A set which contains all forbidden cutting patterns of level r.
z^*	:	Denotes the value of the best known solution.
z_c	:	The optimal value of LP relaxation of E.
β	:	A parameter to control the computation of LP bounds.

Branch and Bound algorithm

1. **Initialization:**
   Set $r := 0, \ s_r := 0, \ E^0 := E, \ b^0 := b, \ z^* := m, \ C := \emptyset, \ V := \emptyset, \ V^1 := \emptyset, \ z(C) := 0$;

2. **Computing actual bounds:**
   If $z(C) + bound_1(E^r) \geq z^*$ then go to Step 3.
   If $s_r \bmod \beta = 0$ and $z(C) + bound_2(E^r) \geq z^*$ then go to Step 3.

If variant (b) and $s_r \bmod \beta \neq 0$ and $z(C) + bound_3(E^r) \geq z^*$ then go to Step 3.
Go to Step 4.

3. **Back track:**
If $r = 0$ then STOP.
If $V^{r+1} \neq \emptyset$ then $V := V \backslash V^{r+1}$, $V^{r+1} := \emptyset$;
$C := C \backslash \{a^r\}$, $V^r := V^r \cup \{a^r\}$, $V := V \cup \{a^r\}$,
$b^r := b^r + a^r$, $z(C) := z(C) - 1$, $r := r - 1$, $s_r := s_r + 1$;
Go to Step 2.

4. **Branching:**
Select the next cutting pattern, say a^*, accordingly to the variants (a) or (b), respectively.
Set $r := r + 1$, $s_r := s_{r-1} + 1$, $a^r := a^*$, $b^r := b^{r-1} - a^r$,
$C := C \cup \{a^r\}$, $z(C) := z(C) + 1$, $V^{r+1} := \emptyset$.
Define E^r.

5. **Applying heuristics:**
Using heuristics 1 and 2 the values z_1 and z_2 are obtained.
If $z(C) + \min\{z_1, z_2\} < z^*$ then $z^* := z(C) + \min\{z_1, z_2\}$.
If $z^* \leq \lceil z_c \rceil$ then STOP;
Go to Step 2.

Remarks: - Only small β-values ($\beta \leq 5$) are tested. Therefore the computation of a new LP bound was not necessary in variant (a) because always cutting patterns were present to be fixed.
- If one is interested only in solutions fulfilling the MIRUP simply the termination test in Step 5 is to modify by "If $z^* \leq \lceil z_c \rceil + 1$ then STOP".

7 Computational results

In order to investigate the one-dimensional cutting stock problem we solved series of randomly generated instances. Thereby the Input-data are chosen from a uniform distribution on some ranges given below. For a given material length L and a chosen $m \in [\underline{m}, \overline{m}]$ the piece lengths l_i are in $[\lceil L/(m-2) \rceil, L/2]$ and the order quantities b_i are in $[2\underline{m}, 10\overline{m}]$.

The LP relaxation for the original problem is solved using the simplex method with column generation where the new pattern is obtained by the greedy algorithm, and if this fails, i.e. the transformed objective function coefficient is nonnegative, the corresponding knapsack problem is solved exactly. Usually a dynamic programming forward state algorithm proves as the best method but also a branch&bound strategy with upper bounds were tested. The latter is of profit if L is large or the order demands b_i are small (in E). Therefore within the branch&bound algorithm the second method is used because of the small order demands in $\overline{E}$. The generation process is terminated if the optimality condition is fulfilled or, secondly, if a given maximum number of solved column generation problems is exceeded or, third, if the decrease of the objective function value is smaller then 0.1 within the last $m/2$ iteration steps.

In the latter two cases it is checked, using the termination criterion, whether the current objective function value z fulfills the condition $\lceil z_c(E) \rceil = \lceil z \rceil$. If not, then the column generation process is continued until one of the termination criteria is fulfilled. The column *ter cri* reports the frequency of these two cases. Hence, $20 - ter\ cri$ is the number how often the LP bound is computed exactly.

The column *val IR* gives the number of instances which have the IRUP.

The columns of *problem sol.* characterize the termination of the residual problem $\overline{E}$ to determine an optimal integer solution. *lb* counts the number of terminations because of Lemma 3 and *h1* and *h2* give the number of instances where heuristic 1 or heuristic 2, respectively, leads to a termination because an optimal solution was found. If these all fail, the branch & bound algorithm must be used and *bb* counts how often this occurs.

In the columns with heading "*average total time*" the average times with respect to 20 instances are reported. The times are given in seconds required on a PC 486 DX, 66 MHz. Here the column $t - LP$ gives the time for solving the LP relaxation of the original problem.

The columns *nodes*, *LP-b* and $t - bb$ give an impression of the complexity of the computed branch&bound-searching-trees. The columns *nodes* contain the minimal and maximal number of the inspected nodes of the searching tree which occur for an instance solved. Similarly, *LP-b* gives the minimal and maximal number of computed LP bounds and $t - bb$ reports all the time required in average per branch&bound computation.

The following tables summarize the results for $L = 1000$ (Tables 2.*), $L = 2000$ (Tables 3.*), $L = 3000$ (Table 4.1), $L = 4000$ (Table 5.1) and $L = 5000$ (Table 6.1). The different L-values are considered to investigate the increase of computational amount because of the dependense of the knapsack algorithms on the size of L. Thereby different values are used for β ($\beta \in \{1, \ldots, 5\}$). For each range $[\underline{m}, \overline{m}]$ 20 instances were generated. The numbers 1 . . . 9 identify the ranges as defined in Table 1.

Table 1 Ranges for the number m of small pieces

range	1	2	3	4	5	6	7	8	9
$\underline{m}$	11	21	31	41	51	61	71	81	91
$\overline{m}$	20	30	40	50	60	70	80	90	100

The tables numbered with N.1 (N $\in \{2, \ldots, 6\}$) contain characteristics for getting an optimal solution. All 900 instances generated could be solved and fulfill the IRUP (column *val IR*). In the most cases an optimal solution was obtained with heuristic 1 (direct greedy method, column *h1*).

Table 2.1 $L = 1000$

no	ter cri	val IR	problem sol. lb	h1	h2	bb	t-LP
1	0	20	0	19	0	1	.3
2	0	20	0	14	0	6	2.5
3	0	20	0	13	1	6	11.0
4	2	20	0	16	2	2	25.3
5	1	20	0	18	0	2	48.2
6	0	20	0	18	0	2	84.6
7	0	20	0	17	0	3	126.4
8	1	20	0	18	0	2	208.5
9	0	20	0	17	0	3	308.5

Table 2.2 $L = 1000$, average total time

no	$\beta = 1$	$\beta = 2$ (a)	$\beta = 2$ (b)	$\beta = 3$ (a)	$\beta = 3$ (b)	$\beta = 4$ (a)	$\beta = 4$ (b)	$\beta = 5$ (a)	$\beta = 5$ (b)
1	.3	.3	.3	.3	.3	.3	.3	.3	.3
2	3.6	3.1	3.2	3.0	3.1	2.9	3.0	2.9	3.0
3	13.7	12.7	12.8	12.3	12.2	12.1	13.0	12.1	13.2
4	29.8	28.5	31.4	28.4	121.9	27.9	123.0	63.3	*26.0
5	53.1	51.3	51.8	50.9	51.3	50.5	51.4	50.3	50.9
6	90.5	88.0	88.5	87.1	93.6	86.9	96.4	86.5	94.7
7	153.0	140.3	141.8	136.4	136.2	134.7	135.9	133.3	134.6
8	230.4	223.7	229.2	226.0	224.3	224.0	226.9	225.2	220.3
9	329.6	319.3	322.6	316.7	319.7	317.7	317.2	317.1	316.1

Table 2.3 $L = 1000$, branch&bound characteristics

		$\beta = 1$			$\beta = 2, (a)$			$\beta = 2, (b)$		
no	bb	nodes	LP-b	t-bb	nodes	LP-b	t-bb	nodes	LP-b	t-bb
1	1	2/2	2/2	.2	3/3	2/2	.3	3/3	2/2	.3
2	6	3/11	3/11	3.5	4/11	2/6	2.0	5/17	3/9	2.3
3	6	3/13	3/13	8.8	3/12	2/6	5.6	5/15	3/8	6.0
4	2	9/20	9/20	37.9	9/19	5/10	26.4	17/44	9/26	54.6
5	2	4/19	4/19	40.9	3/19	2/10	23.4	11/22	6/11	27.1
6	2	11/21	11/21	58.5	14/24	7/12	35.6	17/29	9/15	37.8
7	3	11/35	11/35	176.8	13/26	7/13	95.2	10/35	5/18	102.3
8	2	5/35	5/35	214.7	5/35	3/18	155.7	36/51	18/26	204.0
9	3	1/20	1/20	140.4	1/17	1/9	80.1	1/38	1/19	94.0

Table 2.4 $L = 1000$, branch&bound characteristics

no	bb	$\beta = 3, (a)$ nodes	LP-b	t-bb	$\beta = 3, (b)$ nodes	LP-b	t-bb	$\beta = 4, (a)$ nodes	LP-b	t-bb
1	1	4/4	2/2	.2	3/3	1/1	.1	4/4	1/1	.1
2	6	3/9	1/3	1.5	7/27	3/8	1.8	3/9	1/3	1.2
3	6	5/13	2/5	4.2	4/13	2/5	4.0	4/16	1/4	3.7
4	2	12/24	4/8	25.5	20/1107	8/455	958.9	12/23	3/6	20.0
5	2	3/22	1/8	19.8	12/27	4/10	22.3	3/23	1/6	16.1
6	2	10/23	4/8	27.2	18/190	6/73	89.2	18/24	5/6	25.6
7	3	9/32	3/11	69.2	8/34	3/12	65.2	18/26	5/7	57.8
8	2	23/40	8/14	178.5	44/51	15/17	154.3	19/49	5/13	158.5
9	3	1/19	1/7	62.5	1/39	1/13	74.3	1/35	1/9	69.2

The tables 2.3 - 2.5 show the typical behavior of a branch&bound algorithm. For some series there are instances which need much more computational effort than the remaining. (A similar behavior could be observed for $L = 2000, \ldots, L = 5000$.) In the cases where the values are signed with one or two asterics (* or **) then these values are with respect to 19 or 18 instances. The remaining one or two instances could not be solved with the chosen parameter combination (β, (a) or (b)) because of computer memory restrictions.

Table 2.5 $L = 1000$, branch&bound characteristics

no	bb	$\beta = 4, (b)$ nodes	LP-b	t-bb	$\beta = 5, (a)$ nodes	LP-b	t-bb	$\beta = 5, (b)$ nodes	LP-b	t-bb
1	1	3/3	1/1	.1	4/4	1/1	.1	3/3	1/1	.1
2	6	5/15	2/5	1.4	3/11	1/3	1.2	5/40	1/9	1.4
3	6	4/16	1/4	4.1	4/16	1/4	3.7	6/48	2/16	4.5
4	2	28/1612	10/559	970.1	15/1098	3/279	373.8	*3/11	*3/11	*9.1
5	2	12/47	3/19	23.9	3/24	1/5	14.1	13/57	3/17	18.2
6	2	21/162	6/60	17.5	13/24	3/5	21.0	21/217	5/72	99.6
7	3	7/57	2/15	63.0	11/35	3/7	48.3	7/90	2/24	54.2
8	2	32/98	8/29	179.9	34/68	7/14	170.8	31/76	7/16	114.4
9	3	1/39	1/10	58.0	1/27	1/6	64.9	1/38	1/8	50.3

Table 3.1 $L = 2000$

no	ter cri	val IR	problem sol. lb	h1	h2	bb	t-LP
1	0	20	0	20	0	0	.4
2	0	20	0	17	0	3	3.8
3	0	20	0	12	2	6	15.3
4	9	20	0	16	0	6	40.7
5	10	20	0	16	0	4	77.5
6	9	20	0	17	0	3	145.7
7	6	20	6	14	0	0	235.9
8	7	20	3	12	0	5	342.3
9	8	20	1	13	0	6	507.7

Table 3.2 $L = 2000$, average total time

no	$\beta = 1$	$\beta = 2$ (a)	$\beta = 2$ (b)	$\beta = 3$ (a)	$\beta = 3$ (b)	$\beta = 4$ (a)	$\beta = 4$ (b)
1	.4	.4	.4	.4	.4	.4	.4
2	4.1	4.0	4.0	4.0	4.0	4.0	4.1
3	18.4	17.0	16.9	16.5	16.5	16.3	16.7
4	50.3	46.6	46.4	45.4	45.3	44.9	45.1
5	91.5	88.1	90.7	87.1	87.9	*84.4	88.9
6	179.9	173.8	174.2	172.5	173.2	169.3	179.9
7	247.6	247.0	247.0	244.9	247.0	244.9	244.9
8	452.4	424.6	420.3	446.3	*398.8	455.6	*387.6
9	676.5	639.0	*606.3	601.9	607.2	748.8	**554.7

Table 4.1 $L = 3000$

no	ter cri	val IR	problem sol. lb	h1	h2	bb	t-LP	average total time $\beta = 1$	$\beta = 2$ (a)	$\beta = 2$ (b)	$\beta = 3$ (a)	$\beta = 3$ (b)
1	0	20	0	16	3	1	.7	.7	.7	.7	.7	.7
2	0	20	0	16	0	4	4.1	4.9	4.5	4.5	4.4	4.5
3	3	20	0	15	0	5	22.0	25.2	24.5	25.0	23.9	24.6
4	17	20	3	14	0	3	50.3	69.7	67.3	74.0	*64.8	89.1
5	14	20	6	11	1	2	100.4	135.8	133.3	133.4	258.8	141.4
6	18	20	0	11	0	9	194.3	280.6	263.8	272.9	362.6	329.6
7	11	20	0	13	0	7	277.1	405.7	379.6	386.6	373.4	*650.2
8	14	20	0	13	1	6	457.0	659.0	663.7	638.7	630.0	*125.7
9	16	20	3	11	0	6	662.0	884.2	801.5	*784.6	789.8	*772.5

In the Tables 5.1 and 6.1 additionally a comparison is given between the use of a forward state algorithm (FSS) and a branch&bound algorithm ($b\&b$) for solving the column generation problems to solve the LP relaxation (Q). (In the cases up to $L = 3000$ the forward state algorithm leads always to better running times in comparison to the branch&bound algorithm.)

Table 5.1 $L = 4000$, $\beta = 1$

no	ter cri	val IR	problem sol. lb	h1	h2	bb	FSS t-LP	FSS time	b&b t-LP	b&b time
1	0	20	0	19	1	0	.6	.6	1.0	1.0
2	2	20	0	18	0	2	6.3	7.3	15.4	16.7
3	1	20	5	10	0	5	20.6	24.2	41.4	45.3
4	13	20	5	9	0	6	47.9	80.5	81.0	132.0
5	14	20	0	15	0	5	115.7	189.2	124.2	234.8
6	14	20	1	14	0	5	197.9	275.1	231.8	372.3
7	19	20	4	12	1	3	331.3	478.8	352.3	508.0
8	14	20	1	7	2	10	533.3	910.2	*512.6	*851.5
9	19	20	0	13	0	7	805.2	1178.7	760.4	1201.2

Table 6.1 $L = 5000$, $\beta = 1$

no	ter cri	val IR	problem sol. lb	h1	h2	bb	FSS t-LP	time	b&b t-LP	time
1	0	20	0	16	1	3	.8	1.0	1.4	1.0
2	0	20	1	16	0	3	4.6	5.2	9.0	10.0
3	3	20	4	10	0	6	22.8	30.9	54.8	67.7
4	17	20	1	16	1	2	64.5	101.4	83.4	167.1
5	15	20	3	11	0	6	*133.9	*205.9	178.0	339.7
6	18	20	6	9	0	5	247.6	282.5	291.3	520.4
7	16	20	4	10	1	9	383.8	699.5	254.3	638.7
8	16	20	2	13	1	4	685.0	1023.1	568.7	845.4
9	13	20	4	12	0	4	1049.3	1516.4	907.6	1196.3

As the columns *ter cri* indicate, the number of LP relaxations (Q) solved up to optimality decreases with an increasing number of small pieces and with the size of L. By this the essential importance of the termination criterion (defined in Section 5) gets apparent.

All randomly generated instances of the 1CSP possess the IRUP. The effort of time to compute the knapsack problems for the LP relaxation (for the generation of cutting patterns) strongly increases with the material length L. This is the effect of the used forward state algorithm within the computation of the initial lower LP bound.

A comparison between the choice of various values of β shows that it is generally of profit to use $\beta > 1$ but the worst-case-behaviour advise to choose $\beta = 2$ or $\beta = 3$. The difference between variants (a) (LP-cutting-pattern-strategy) and (b) (high-density-cutting-pattern-strategy) is not of an important significance in relation to the LP relaxation of the original problem. The variant (a) is better if the residual problems $\overline{E}$ are relatively big and the other variant is of profit if the order demands of $\overline{E}$ are very small.

Additionally it is to remark, if the computation should be terminated when a solution is found proving the MIRUP then for all instances the application of the branch&bound algorithm was not necessary (in similarity to [13]).

8 Conclusional Remarks

In this paper we proposed a branch & bound algorithm for the one-dimensional cutting stock problem. The essential feature of this algorithm is the orientation on the conjecture that the modified integer round-up property holds for the problem considered.

The computational experiments with problems with up to 100 small pieces show that branching is necessary only in about 20% of the instances. Proportional to the increase of the material length L the computational time increases too because the solution of the knapsack problems in the column generation process needs more time.

The average computational time shows that the proposed algorithm is a very good tool for solving instances of the one-dimensional cutting stock problem exactly.

Acknowledgement

The authors wish to thank Uta Sommerweiß for implementing the algorithms and doing the extensive computational tests.

References

[1] S. Baum and L.E. Trotter, Jr., Integer rounding for polymatroid and branching optimization problems, SIAM J. Alg. Disc. Meth. 2 (1981) 4, 416-425.

[2] E.G. Coffmann,jr., M.R. Garey, D.S. Johnson, R.E. Targon, Performance bounds for level oriented two-dimensional packing algorithms, SIAM J.Comput. 9 (1980)p.808-826.

[3] A. Diegel, Integer LP solution for large trim problem, Working Paper, University of Natal, South Africa, 1988.

[4] H.Dyckhoff, and U.Finke, Cutting and packing in production and distribution, Physica Verlag, Heidelberg, 1992.

[5] M. Fieldhouse, The duality gap in trim problems, SICUP-Bulletin No. 5, 1990.

[6] P.C. Gilmore and R.E. Gomory, A linear programming approach to the cutting stock problem, Operations Res. 9 (1961) 849-859.

[7] P.C. Gilmore and R.E. Gomory, A linear programming approach to the cutting stock problem - Part II, Operations Res. 11 (1963) 863-888.

[8] R.E. Johnston, Rounding algorithms for cutting stock problems, Asia-Pacific J. of OR 3 (1986) 166-171.

[9] O. Marcotte, The cutting stock problem and integer rounding, Mathematical Programming 33 (1985) 82-92.

[10] O. Marcotte, An instance of the cutting stock problem for which the rounding property does not hold, Oper. Res. Lett. 4 (1986) 5, 239-243.

[11] G.L.Nemhauser and L.A.Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons, New York 1988.

[12] G. Scheithauer and J. Terno, About the gap between the optimal values of the integer and continuous relaxation one-dimensional cutting stock problem, *in* Operations Research Proceedings 1991, pp. 439-444, Springer-Verlag, Berlin, Heidelberg, 1992.

[13] G. Scheithauer and J. Terno, The modified integer round-up property for the one-dimensional cutting stock problem, Preprint MATH-NM-10-1993, TU Dresden (submitted for publication).

[14] G. Scheithauer and J. Terno, Theoretical investigations on the modified integer round-up property for one-dimensional cutting stock problem, Preprint MATH-NM-12-1993, TU Dresden (submitted for publication).

[15] G. Scheithauer and J. Terno, Equivalence of cutting stock problems, Working Paper, TU Dresden, 1993.

[16] J. Terno, R. Lindemann und G. Scheithauer, Zuschnittprobleme und ihre praktische Lösung, Verlag Harry Deutsch, Thun und Frankfurt/Main, und Fachbuchverlag Leipzig 1987.

[17] G.Wäscher and T.Gau, Two approaches to the cutting stock problem, IFORS '93 Conference, Lisboa 1993.

Author's address:
Prof. Dr. Johannes Terno, Dr. Guntram Scheithauer, Institute of Numerical Mathematics, Technical University Dresden,
Mommsenstr. 13, D - 01062 Dresden, Germany

E-mail: terno@math.tu-dresden.de, scheit@math.tu-dresden.de