Cutting Stock with Bounded Open Stacks: a New Integer Linear Programming Model

Claudio Arbib, Fabrizio Marinelli, Paolo Ventura

Technical Report TRCS 007/2010

Dipartimento di Informatica
Università degli Studi dell'Aquila
Via Vetoio Loc. Coppito
I-67010 L'Aquila, Italy

http://www.di.univaq.it

Technical Report TRCS Series

Cutting Stock with Bounded Open Stacks: a New Integer Linear Programming Model

Claudio Arbib
Dipartimento di Informatica
Università degli Studi dell'Aquila
via Vetoio, Coppito, I-67010 L'Aquila, Italy
e-mail: claudio.arbib@univaq.it

Fabrizio Marinelli
Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione
Università Politecnica delle Marche
via Brecce Bianche 12, I-60121 Ancona, Italy
e-mail: marinelli@diiga.univpm.it

Paolo Ventura
Istituto di Analisi dei Sistemi e Informatica
Consiglio Nazionale delle Ricerche
viale Manzoni 30, I-00185 Roma, Italy
e-mail: paolo.ventura@iasi.cnr.it

Abstract

We address a 1-dimensional cutting stock problem where, in addition to trim-loss minimization, we require that the set of cutting patterns forming the solution can be sequenced so that the number of stacks of parts maintained open throughout the process never exceeds a given s. For this problem, we propose a new integer linear programming formulation whose constraints grow quadratically with the number of distinct part types.

1 Introduction

The 1-dimensional *Cutting Stock Problem* (1-CSP, see [5] for an old but vaste annotated bibliography) calls for cutting d_i parts w_i units wide ($i \in B$, $|B| = m$) from a minimum number of identical stock items of width w. A *cutting pattern* specifies a way to cut parts from a stock item: in the 1-dimensional case, it corresponds to an integer solution $\mathbf{p} = (p_1, \ldots, p_m) \geq 0$ of inequality $\sum_{i \in B} w_i p_i \leq w$ (knapsack condition). A solution of the 1-CSP provides (i) a set P of distinct patterns, (ii) the number of times each pattern of P must be repeated in order to cover the parts requirement: this number is called the pattern *run length*. In practice, to reduce setups it is customary to non-preemptively run each cutting pattern up to its run length. The sequence in which patterns are used, in fact a permutation π of P, together with P gives a *cutting plan* (P, π).

Alongwith the classical objective of trim-loss minimization, a major goal of industries deploying a cutting process is to organize production in a way compliant with machine-tool resources and capacity. Under this respect, an important indicator is the number of *open stacks*: after cut, parts of the same size are normally collected into a stack. A stack is open (closed) as soon as the first (the last) of its parts has been cut. Because automated cutting machines usually have a downstream buffer with limited capacity, only a limited number s of stacks can be maintained open throughout the cutting process.

In the literature, a non-negligible amount of research can be found on the *Open Stack Minimization Problem*, known with the acronym MOSP, which calls for permuting a given set of patterns so as to minimize the maximum number of open stacks. Solving the MOSP was the subject of an international contest: in [9], Smith and Gent collected a quite large number of papers with algorithm details and computation experiments. Recent research focussed on 0-1 matrices with the *Consecutive Ones Property* (C1P) and provided an interesting integer linear programming formulation using Oswald-Reinelt inequalities [4]; but a breakthrough [3] has recently been achieved by a smart combinatorial lower bound that, used in a branch-and-bound scheme, reduced by a factor of 10^{-6} the best computation times in [9].

Apparently, the MOSP can be applied within a decomposition approach where one first finds a set of patterns minimizing trim-loss, and then sequences those patterns minimizing the maximum number of open stacks. Unfortunately, this decomposition has a couple of important drawbacks:

- first, it is not the same as minimizing open stacks under a prescribed trim-loss: in fact, cutting plans with similar trim-losses may be obtained by many different pattern sets, and the quality of a pattern sequence (and the time needed to compute it) can strongly depend on the number and type of patterns chosen;

- moreover, even an optimal solution to the lattter problem may require more open stacks than allowed by the cutting machine, and then turn out to be practically infeasible.

The correct way to address the problem is in fact to finding a pattern set that both covers demand and can be sequenced without exceeding buffer capacity: doing that with the goal of minimizing trim-loss is what we here call 1-*Dimensional Cutting Stock with Bounded Open Stacks Problem* (1-CS-BOSP).

To the best of our knowledge, the 1-CS-BOSP has so far registered two contributions. Yanasse and Pinto Lamosa [11] formulate the problem in terms of integer linear programming, but the formulation, with exponentially many variables and constraints, is far from being practical and just helps develop a lower bound (by lagrangian relaxation) and a primal heuristic. Belov and Scheithauer [2] propose an effective sequential heuristic, that makes use of pseudo-prices in order to select the part types to be cut pattern after pattern.

In this paper we propose a new integer linear programming formulation which, unlike Yanasse and Pinto Lamosa's, is compact in the number of constraints. We start from the well-known Kantorovich's model for the 1-CSP, and introduce variables in order to count (and bound) the number of open stacks. This formulation has a pseudo-polynomial number of variables and activation constraints. After convexifying knapsack constraints, we see that the number of non-trivial inequalities of the resulting integer linear programming formulation is $O(m^2)$.

2 Formulating the problem

Let us formally state the 1-CS-BOSP as follows:

Problem 2.1 *Given*

- *an unlimited number of stock items of width w,*
- *a set of part types B of distinct widths w_i, required in amounts d_i, $i \in B$,*
- *a positive integer s,*

find cutting plan (P, π) with run lengths that exactly cover the demand $\mathbf{d}$ and such that

- *the number of open stacks never exceeds s,*
- *the total number of stock items used (= sum of run lengths) is minimized.*

Yanasse and Pinto Lamosa [11] formulate Problem 2.1 by merging Gilmore-Gomory's model and a variant of Tang-Denardo's tool-switch formulation [10]. The resulting model:

- explicitly generates patterns $\mathbf{a}_k$, the 0-1 incidence vectors of the part types produced by each pattern, and integer-valued run lengths y_k;
- uses assignment variables x_{kn} stating whether pattern k is or not sequenced by the n-th; 0-1 variables u_{in} stating whether the n-th pattern feeds or not the i-th stack with a part type; 0-1 variables v_{in} stating whether the i-th stack is or is not still open between the n-th and the $(n+1)$-st pattern;
- links run lengths and assignment variables via 0-1 activation variables v_k that take value 1 if and only if $y_k > 0$.

Apparently, the formulation is independent on the dimension of parts, but complexity of course increases in the 2-dimensional case. Candidate patterns are in principle exponentially many in m, and therefore so is the number of both variables and constraints. In detail, the Authors count $p^2 + 2pm + 2p - m$ variables and

$p^2m + pm + 5p + 1$ constraints for p patterns, and observe that these numbers are quite large even for small values of m: for instance, $m = 20$ and $p = 1000$ already give $1,041,980$ variables and $20,025,001$ constraints, a number that can prevent from solving even small problems.

In our formulation, we initially suppose that a stock item is cut at every time unit. To give a first formulation of Problem 2.1 we suppose to know an upper bound u to the number of stock items cut in an optimal solution. In the 1-dimensional case an upper bound on the number of parts of type i that can be cut from a stock item trivially is $M_i = \lfloor \frac{w}{w_i} \rfloor$. Let us define the following decision variables for $i \in B, j = 1, \ldots, u$:

- x_{ij} = how many parts of type i are cut from the j-th stock item, $x_{ij} \leq M_i$;
- $y_j = 1$ if the j-th stock item is at all cut, else 0;
- $q_{ij} = 1$ if the i-th stack is open at some time $\leq j$, else 0;
- $r_{ij} = 1$ if the i-th stack is closed at some time $\leq j$, else 0.

Variables x_{ij}, y_j are the same as in Kantorovich's model [8]. Variables q_{ij}, r_{ij}, introduced to count open stacks, play a role similar to x_{in} in [11].

$$\min \sum_{j=1}^{u} y_j \tag{1}$$

$$\sum_{j=1}^{u} x_{ij} = d_i \qquad i \in B \tag{2}$$

$$\sum_{i \in B} w_i x_{ij} \leq w y_j \qquad i \in B, 1 \leq j \leq u \tag{3}$$

$$x_{ij} \leq M_i(q_{ij} - r_{ij}) \qquad i \in B, 1 \leq j \leq u \tag{4}$$

$$q_{ij} \leq q_{i,j+1} \qquad i \in B, 1 \leq j < u \tag{5}$$

$$r_{ij} \leq r_{i,j+1} \qquad i \in B, 1 \leq j < u \tag{6}$$

$$\sum_{i \in B} (q_{ij} - r_{ij}) \leq s \qquad 1 \leq j \leq u \tag{7}$$

$$x_{ij} \in \mathbb{N} \qquad i \in B, 1 \leq j \leq u \tag{8}$$

$$y_j, q_{ij}, r_{ij} \in \{0, 1\} \qquad i \in B, 1 \leq j \leq u \tag{9}$$

Note that $x_{ij} \geq 0$ and inequality (4) give $r_{ij} \leq q_{ij}$ for all i and j: that is, no stack can be closed if it has not been opened before.

The upper bound u can be taken as large as the number of stock items cut in any feasible solution: trivially, using only patterns that produce a single part type, $u = \sum_{i \in B} \lceil \frac{d_i}{\lfloor w/w_i \rfloor} \rceil$ cuts are sufficient to cover demand opening one stack at a time.

The number of constraints involving variables q_{ij}, r_{ij} can be reduced by reducing the number of these variables. Indeed, one can group the cut sequence by time intervals, and then refer q, r variables and inequalities (5)-(7) to the h-th interval instead of the j-th cut. With J_h denoting the set of cuts operated in interval h, inequality (4) s replaced by

$$\sum_{j \in J_h} x_{ij} \leq d_i(q_{ij} - r_{ij}) \qquad\qquad i \in B, 1 \leq j \leq u \tag{10}$$

Although reduced, the number of variables and constraints of model (1)-(9) is still pseudo-polynomial and, as such, can happen to be too large in practical cases. Another well known drawback of model (1)-(9) is that variable values can be exchanged without changing the solution, i.e., the model has a strong symmetry.

3 Reformulating the problem

Let $\mathbf{p}^k = (p_i^k)_{i=1}^m \geq 0$ denote a cutting pattern (i.e., an integer solution of knapsack condition $\sum_{i \in B} w_i x_i \leq w$), $P = \{\mathbf{p}^k\}_{k \in K}$ be a set of patterns, and $\mathbf{T}$ a 0-1 matrix with $m = |B|$ rows, none of which null. We call $\mathbf{T}$ a *track* for P if, for any $k \in K$, there exists a column h with $t_{ih} > 0$ for all $i \in B$ such that $p_i^k > 0$. Reciprocally, we say that P is *supported* by $\mathbf{T}$, and in particular that a single part type i is supported by a column $\mathbf{t}_h$ of $\mathbf{T}$ if $t_{ih} = 1$. With no loss of generality, from now on we assume the columns of $\mathbf{T}$ mutually non-dominated, that is, no two columns $\mathbf{t}_j, \mathbf{t}_h$ are such that $t_{ij} \leq t_{ih}$ for all $i \in B$.

Let $\omega(\mathbf{T})$ denote the largest number of non-zero elements in a column of a track $\mathbf{T}$. We then say that $\mathbf{T}$ is *feasible* if it has the *consecutive one property* (C1P) by rows, and $\omega(\mathbf{T}) \leq s$. From now on, suppose the columns of $\mathbf{T}$ ordered according to the C1P by rows.

Proposition 3.1 *A set P of cutting patterns is schedulable if and only if it is supported by a feasible track $\mathbf{T}$.*

Proof. Suppose P schedulable in an order $\pi = (1, \ldots, n)$, and let $\bar{\mathbf{T}}$ be an $m \times n$ 0-1 matrix such that $\bar{t}_{ik} = 1$ if and only if the k-th pattern of the order produces parts of the i-th batch. Fill then each row of $\bar{\mathbf{T}}$ with the least number of 1's required to give it the C1P, remove dominated columns and call $\mathbf{T}$ the resulting matrix, which clearly supports P. Now every 0 replaced by a 1 corresponds to a part type not yet completed, i.e., to an open stack, and since π maintains open $\leq s$ stacks, every column of $\mathbf{T}$ has no more than s non-zero elements: therefore, $\mathbf{T}$ is feasible.

Conversely, let $\mathbf{T}$ be a 0-1 matrix with the C1P and $\omega(\mathbf{T}) \leq s$, and let P be a set of patterns supported by $\mathbf{T}$, that is, for any pattern of P there exists an index h such that $t_{ih} = 1$ if part type i is produced by that pattern. Since the columns of $\mathbf{T}$ are mutually non-dominated, at least one part type supported by column h, say i, is not supported by column $h + 1$ (and vice-versa). Since $\mathbf{T}$ has the C1P, once a part type is no longer supported by a column, it is not supported by any of the subsequent ones. So no pattern of P following those supported by the h-th column produces part type i, and since no more than s part types are supported by any column, P is schedulable. □

By Proposition 3.1, Problem 2.1 is restated without mentioning pattern sequencing:

Problem 3.1 *Find a set P of cutting patterns that produces all the parts required with a minimum trim loss and is supported by a feasible track.*

We now exploit the idea of track in a new formulation. Let $A = \{\mathbf{p}^1, \ldots, \mathbf{p}^p\} \subseteq \mathbb{N}^m$ denote the set of all feasible solutions to knapsack condition. Using $\lambda_j^k \in \{0,1\}$ to select the k-th element of A, write

$$\mathbf{x}_j = \sum_{k=1}^{p} \mathbf{p}^k \lambda_j^k \tag{11}$$

with $\sum_k \lambda_j^k = 1, \lambda_j^k \geq 0$ and integer. Replacing x_{ij} by (11) in (1)-(3), (9), and setting $z^k = \sum_j \lambda_j^k$ we get Gilmore-Gomory's model

$$\begin{aligned} \min \sum_{k=1}^{p} z^k & & & & (12)\\ \sum_{k=1}^{p} p_i^k z^k &= d_i & & i \in B \\ z^k &\in \mathbb{N} & k = 1, \ldots, p & \end{aligned}$$

which, substituting exponentially many variables for pseudo-polynomially many constraints of type (3), is well known to perform much better than Kantorovich's. Let $\mathbf{T}$ be an $m \times n$ feasible track, K_h contain the indexes of patterns $\mathbf{p}^k$ supported by the h-th column of $\mathbf{T}$, and z_h^k denote the run length of the k-th pattern of K_h. Plug $z^k = \sum_{h=1}^{n} z_h^k$ into (12), and note that variables q_{ih} and r_{ih} constrained as in (5), (6), (10) give $t_{ih} = q_{ih} - r_{ih}$. These variables can therefore be used to reformulate Problem 2.1:

$$\min \sum_{h=1}^{n} \sum_{k \in K_h} z_h^k \tag{13}$$

$$\sum_{h=1}^{n} \sum_{k \in K_h} p_i^k z_h^k = d_i \qquad i \in B \tag{14}$$

$$\sum_{k \in K_h} p_i^k z_h^k \leq d_i(q_{ih} - r_{ih}) \qquad i \in B, 1 \leq h \leq n \tag{15}$$

$$\sum_{i=1}^{m} (q_{ih} - r_{ih}) \leq s \qquad 1 \leq h \leq n \tag{16}$$

$$q_{ih} - q_{i,h+1} \leq 0 \qquad i \in B, 1 \leq h < n \tag{17}$$

$$r_{ih} - r_{i,h+1} \leq 0 \qquad i \in B, 1 \leq h < n \tag{18}$$

$$z_h^k \in \mathbb{N} \qquad 1 \leq h \leq n, 1 \leq k \leq p \tag{19}$$

$$q_{ih}, r_{ih} \in \{0,1\} \qquad i \in B, 1 \leq h \leq n \tag{20}$$

It is easy to see that model (13)-(20) derives from Dantzig-Wolfe decomposition of integer program (1)-(3),(5)-(9),(10) rewritten with q_{ih}, r_{ih}. Compactness of the new model depends on n. To evaluate it, it is useful to point out dominance properties among tracks.

Lemma 3.2 *Every feasible track* $\mathbf{T}$ *with* $\omega(\mathbf{T}) = \omega$ *is dominated by a feasible track with exactly* ω *non-zero elements per column.*

Proof. Let C be a subset of consecutive columns of $\mathbf{T}$ having $< \omega$ non-zero elements. Clearly, C does not consist of all the columns of $\mathbf{T}$, otherwise $\omega(\mathbf{T}) < \omega$. Thus, w.l.o.g., a column $\mathbf{t}$ with exactly ω non-zero elements follows the rightmost column, say $\mathbf{v}$, of C. Since by general assumption $\mathbf{v}$ and $\mathbf{t}$ do not dominate each other, there are at least two rows i, j such that $v_i < t_i$, $v_j < t_j$. In other words, as $\mathbf{T}$ has the C1P, t_i and t_j are the leftmost 1's of rows i and j, and therefore one can replace v_i by 1 preserving the C1P and mutual non-dominance, so augmenting the non-zeroes of $\mathbf{v}$. Repeating the argument one gets the thesis. □

Lemma 3.3 *Every feasible track* $\mathbf{T}$ *with* $\omega(\mathbf{T}) = s$ *is dominated by a feasible track where any two adjacent columns differ by exactly two elements.*

Proof. Suppose that consecutive columns $\mathbf{v}, \mathbf{t}$ differ by more than two elements: w.l.o.g. suppose $v_i > t_i$ for $i = 1, \ldots, h$ and $v_i < t_i$ for $i = h+1, \ldots, 2h$, for some $h > 1, h \leq s$. Then we can interpose new columns $\mathbf{v}_j$ between $\mathbf{v} = \mathbf{v}_1$ and $\mathbf{t}$, $1 < j \leq h$, in such a way that

$$\begin{cases} v_{ij} > v_{i,j+1} & \text{for } i = j \\ v_{ij} < v_{i,j+1} & \text{for } i = j+h \\ v_{ij} = v_{i,j+1} & \text{otherwise} \end{cases}$$

The resulting track clearly has the C1P and supports any cutting plan supported by $\mathbf{T}$. □

We then conclude

Theorem 3.4 *The number of constraints* (14)-(18) *goes with* m^2*.*

Proof. In fact, Propositions 3.2 and 3.3 imply that the relevant tracks have $n = m - s + 1$ columns. □

With $m = 20$ and $s = 4$ the number of constraints becomes $1,017$, dropping that in [11] by almost $20,000$ times.

4 Conclusions and future research

We considered a problem of stock cutting where cuts must be sequenced so that the number of distinct part types produced at any time (open stacks) does not exceed a prescribed s. The problem was discussed referring to the recent literature on cutting and sequencing problems, and then formulated as integer linear programming. Like the formulation given in [11], the one proposed has exponentially many variables, but unlike it, it has much less constraints (in fact, quadratically many in the number of part types). Further research is to be done in order to state its computational features.

References

[1] Arbib, C., and F. Marinelli, A Heuristic Algorithm to Schedule Batches in 1-Dimensional Stock Cutting with Limited Downstream Buffer, in preparation

[2] Belov, G., and G. Scheithauer, Setup and Open-Stacks Minimization in One-Dimensional Stock Cutting, *INFORMS J. on Computing* **19**, 1 (2007) 27-35

[3] Chu, G. G., and P. Stuckey, Minimizing the Maximum Number of Open Stacks by Customer Search, *Lecture Notes in Computer Science: Principles and Practice of Constraint Programming CP2009*, Springer-Verlag, Vol. 5202 (2009) 242-257.

[4] De Giovanni, L., F. Pezzella, M. Pfetsch, G. Rinaldi, and P. Ventura, The Open Stack Problem, *Giornate di Lavoro AIRO*, Genova, Italy (September 6, 2007)

[5] Dyckhoff, H., G. Scheithauer, and J. Terno, Cutting and Packing: An Annotated Bibliography, `ibl@ibl.RWTH-aachen.de` (1996)

[6] Gilmore, P.C, and R.E. Gomory, A Linear Programming Approach to the Cutting Stock Problem, *Operations Research* **8** (1961) 849-859

[7] Gilmore, P.C., and R.E. Gomory, A Linear Programming Approach to the Cutting Stock Problem – Part II, *Operations Research* **11** (1963) 863-888.

[8] Kantorovich, L.V., Mathematical Models of Organising and Planning Production, *Management Science* **6** (1960) 366-422

[9] Smith, B.M., and I.P. Gent, Constraint Modelling Challenge 2005, *Fifth Workshop on Modelling and Solving Problems with Constraints*, IJCAI 2005, Edinburgh, Scotland (July 31, 2005)

[10] Tang, C.S., and E.V. Denardo, Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches, *Operations Research* **36** (1988) 767-777.

[11] Yanasse, H.H., and M.J. Pinto Lamosa An integrated cutting stock and sequencing problem, *European Journal of Operational Research* **183** (2007) 1353-1370