# A Class of Subpattern Formulations for One-Dimensional Stock Cutting

Gleb Belov[§¶] Robert Weismantel[‖]

*November 18, 2003*

**Abstract**

In the classical Gilmore-Gomory model for the *one-dimensional cutting stock problem* (**1D-CSP**) we have to deal implicitly with a huge number of variables representing all feasible patterns. The advantage is a strong relaxation and absence of symmetries. To reduce the number of variables, we can restrict them to *subpatterns*, i.e., partial patterns which are combined to produce whole patterns. Each subpattern can be a part of different resulting patterns; thus, all patterns have to be numbered, which brings much symmetry into the model. Moreover, the continuous relaxation may not only have non-integer pattern application intensities but also overcapacity patterns, resulting in a weak bound. However, it is possible to choose a small set of elementary subpatterns so that standard optimization software can be applied to the model.

**Keywords:** cutting, symmetries

## 1 Introduction

The one-dimensional cutting stock problem (1D-CSP) is defined by the following data: ($m$, $L$, $l = (l_1, \ldots, l_m)$, $b = (b_1, \ldots, b_m)$), where $L$ denotes the length of each stock piece, $m$ denotes the number of smaller piece types and for each type $i = 1, \ldots, m$, $l_i$ is the piece length, and $b_i$ is the order demand. In a *cutting plan* we must obtain the required set of pieces from the available stock lengths. The objective is to minimize the number of used stock lengths or, equivalently, trim loss (waste). In a real-life cutting process there are some further criteria, e.g., the number of different cutting patterns (setups) and open stacks [BS03].

In 1961, Gilmore and Gomory [GG61] proposed the following model of 1D-CSP. A *cutting pattern* describes how many items of each type are cut from a stock

[§]Dresden University, Institute of Numerical Mathematics, e-mail: *belov@math.tu-dresden.de*.

[¶]Research was supported by the DAIMLER-BENZ Foundation.

[‖]Otto-von-Guericke-University of Magdeburg, Department of Mathematics, e-mail: *robert.weismantel@mathematik.uni-magdeburg.de*.

length. Let column vectors $a^j = (a_{1j}, \ldots, a_{mj}) \in \mathbb{Z}_+^m$, $j = 1, \ldots, n$, represent all possible cutting patterns. To be a valid cutting pattern, $a^j$ must satisfy

$$\sum_{i=1}^{m} l_i a_{ij} \leq L \tag{1}$$

(knapsack condition). Moreover, we consider only *proper* patterns:

$$a_{ij} \leq b_i, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \tag{2}$$

because this reduces the search space of the continuous relaxation in instances where the demands $b$ are small [NST99].

Let $x_j$, $j = 1, \ldots, n$, be the *frequencies* (*intensities, multiplicities, activities*, i.e., the numbers of application) of the patterns in the solution. The model of Gilmore and Gomory is as follows:

$$z_{\text{G\&G}}^{\text{1D-CSP}} = \min \sum_{j=1}^{n} x_j \tag{3}$$

$$\text{s.t.} \qquad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m \tag{4}$$

$$x_j \in \mathbb{Z}_+, \quad j = 1, \ldots, n. \tag{5}$$

The huge number of variables/columns is not available explicitly for practical problems. Usually, necessary patterns are generated during a solution process (*column generation*, *pricing*). However, the number of different patterns in a solution cannot be greater than the number of stock lengths and is usually comparable with the number of piece types.

The model has a very strong relaxation. There exists the conjecture *Modified Integer Round-Up Property* (**MIRUP**, [ST95]):

> *The gap between the optimum value $z_{G\&G}^{1D\text{-}CSP}$ and the optimum relaxation value $\underline{z}_{G\&G}^{1D\text{-}CSP}$ (obtained by allowing non-integer variable values) is always smaller than 2.*

Actually, there is no instance known with a gap greater than 7/6 [RS02]. Moreover, instances with a gap smaller than 1 constitute the vast majority. These are called *Integer Round-Up Property* (**IRUP**) instances. Instances with a gap greater than or equal to 1 are called **non-IRUP** instances.

The model has a huge number of variables. But no variable values can be exchanged without changing the solution, i.e., the model has no symmetry. This and the strength of the relaxation make it advantageous for an enumerative approach. In the survey [dC02] we find several alternative ILP models of 1D-CSP. Kantorovich [Kan60] proposed an assignment formulation. Dyckhoff [Dyc81] and Stadtler [Sta88] discussed *one-cut* models. Later, Valério de Carvalho [dC98] proposed an *arc flow formulation*. Similar to one-cut models, it considers physical

positions of items in the patterns. Both one-cut and arc flow models are equivalent to the Gilmore-Gomory formulation: solutions given by network flows can be decomposed into paths representing patterns. The number of the flow variables is pseudo-polynomial. The assignment model has a very weak continuous relaxation (cf. [VBJN94]) and much symmetry.

# 2 Subpattern Formulations

Let $\overline{\mathcal{A}} = \{\overline{a}^j\}$ ($j = 1, \ldots, \eta$) be a set of subpatterns such that each feasible pattern $a$ can be represented as a sum of elements of $\overline{\mathcal{A}}$. For example, we could define $\overline{\mathcal{A}}$ as a set of half-patterns so that at most two are needed in a combination:

$$\left\{\overline{a} \in \mathbb{Z}_+^m : \; l\overline{a} \in \left(\frac{L - l_{\min}}{2} - \frac{l_{\max}}{2}, \frac{L}{2} + \frac{l_{\max}}{2}\right]\right\} \tag{6}$$

In general case, we may demand that each pattern is combined of maximally $n^p$ subpatterns, $n^p > 2$:

$$\left\{\overline{a} \in \mathbb{Z}_+^m : \; l\overline{a} \in \left(\frac{L - l_{\min}}{n^p} - l_{\max}, \frac{L}{n^p} + l_{\max}\right]\right\} \tag{7}$$

The advantage of taking large $n^p$ appears when all pieces are small. For a pattern containing several items of the same type, it may be possible that all or several subpatterns are equal so that the set of subpatterns has to contain up to $n^p$ equal single-product subpatterns (see the model below).

Note that when using half-patterns (6) or $n^p$-patterns (7), we are guaranteed only *maximal* resulting patterns, i.e., $\{a \in \mathbb{Z}_+^m : \; 0 \le L - la < l_{\min}\}$. Thus, it is necessary to allow overproduction of items, so that the total amount of items of type $i$ in the solution may be larger than $b_i$. Moreover, we cannot demand that the resulting patterns be proper ($a \le b$).

Alternatively, we may use the following set of single-product subpatterns:

$$\left\{2^p e^i : \; i = 1, \ldots, m, \; p = 1, \ldots, \lfloor \log_2(\min\{b_i, L/l_i\}\rfloor\right\} \tag{8}$$

($e^i$ is the $i$-th unit vector). In (8), if we want the resulting patterns to be proper ($a \le b$), we need additional constraints, although this is done implicitly in the standard model.

Let $\overline{K}$ be an upper bound on the number of different patterns in an optimum solution. Let $x_{kj}$ be the frequency with which subpattern $\overline{a}^j$ is used in the solution as a part of pattern $k$ ($k = 1, \ldots, \overline{K}$, $j = 1, \ldots, \eta$). Let $y_{kj} = 1$ if $\overline{a}^j$ is a part of pattern $k$, otherwise $y_{kj} = 0$. Then $x_{kj} \le x_j^{\max} y_{kj}$, $\forall k, j$, where

$$x_j^{\max} = \min_{i:\overline{a}_{ij} > 0} \{\lfloor b_i / \overline{a}_{ij} \rfloor\}, \quad \forall j \tag{9}$$

is an upper bound on the intensity of subpattern $j$ in a solution. Let $q_k$ be the intensity of pattern $k$: $q_k \geq x_{jk}, \forall j, k$. The model is as follows:

$$\min \textstyle\sum_k q_k \tag{10}$$

$$\text{[SUBPAT]} \qquad \text{s.t.} \quad \textstyle\sum_k \sum_j \overline{a}_{ij} x_{kj} \geq b_i, \qquad \forall i \tag{11}$$

$$x_{kj} \leq x_j^{\max} y_{kj}, \quad \forall k, j \tag{12}$$

$$\textstyle\sum_j l\overline{a}^j y_{kj} \leq L, \qquad \forall k \tag{13}$$

$$q_k \geq x_{kj}, \qquad \forall k, j \tag{14}$$

$$x_{kj}, q_k \in \mathbb{Z}_+, \qquad \forall k, j \tag{15}$$

$$y_{kj} \in \mathbb{B}, \qquad \forall k, j, \tag{16}$$

where (13) restricts the length of any resulting pattern.

Note that we allow a subpattern to be in a pattern not more than once (16) as otherwise we cannot linearly measure $q_k$. The same reason applies to having a separate variable $x_{kj}$ for each $k$. Furthermore, the positive components $x_{kj}$ for pattern $k$ may have different values for different $j$. This creates several actual patterns, all of which satisfy (13), but the model 'knows' only about pattern $k$. Thus, the total actual number of patterns in a solution may be larger than $\overline{K}$.

In the solution of the continuous relaxation, (12) may allow that $y_{kj} < 1$, which leads to overcapacity patterns. The number of the constraints (12) and (14) is $\overline{K}\eta$, which is the number of $x$-variables. Thus, we have to choose a set $\overline{\mathcal{A}}$ with a sensible number of subpatterns or apply column generation.

To strengthen the relaxation, we may add the constraint

$$\textstyle\sum_j l\overline{a}^j y_{kj} \leq L q_k, \qquad \forall k, \tag{17}$$

cf. (13). In $n^p$-subpatterns (6) and (7),

$$\textstyle\sum_j y_{kj} \leq n^p, \qquad \forall k$$

can also strengthen the relaxation. For single-product subpatterns, to restrict the resulting patterns to be proper, we need

$$\textstyle\sum_j \overline{a}^j y_{kj} \leq b, \qquad \forall k. \tag{18}$$

But the relaxation is weak:

**Example.** Consider the instance ($m = 3$, $L = 30$, $l = (6, 10, 15)$, $b = (100, 100, 100)$). An optimum can be easily constructed: $20 \times (5, 0, 0)$, $33 \times (0, 3, 0)$, $50 \times (0, 0, 2)$, $1 \times (0, 1, 0)$. Starting with this solution in the relaxation

of (10)–(18) using subpatterns (8):

$$\overline{\mathcal{A}} = \begin{pmatrix} 1 & 2 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}$$

$$\begin{array}{lccccccll} x_{1j} = & 20 & & 20 & & & & & q_1 & = 20 \\ x_{2j} = & & & & 33 & 33 & & & q_2 & = 33 \\ x_{3j} = & & & & & & & 50 & q_3 & = 50 \\ x_{4j} = & & & & 1 & & & & q_4 & = 1, \end{array}$$

it is easy to come to a better solution by allowing 7 items of type 1 in the first pattern. Set $x_{11} = x_{12} = x_{13} = q_1 = 14\frac{2}{7}$ and $y_{11} = 0.15$, $y_{12} = 0.3$, $y_{13} = 0.6$ so that (12) is not violated. Hence (13) is fulfilled: $6(1 \cdot 0.15 + 2 \cdot 0.3 + 4 \cdot 0.6) = 6 \cdot 3.15 < 20 < L = 30$. The LP value is smaller by more than 5 units. A further reduction will be achieved by incorporating the subpattern (0,0,1), i.e., by allowing 3 items of type 3 in the third pattern.

However, with the single-product subpatterns (8) it is obvious that no column generation is needed. Thus, standard optimizers can be applied. Anti-symmetry constraints like $q_k \geq q_{k+1}$, $k = 1, \ldots, \overline{K} - 1$, can reduce the search space. One open question in the subpattern models concerns the number of different patterns $\overline{K}$. In [BS03] it was always possible to find solutions with the minimum material input and $\overline{K} \leq m$. This could change if we modify the model to minimize the number of open stacks as in [BS03].

Various formulations of 1D-CSP and of other IP problems are sometimes used to derive branching rules for LP-based branch-and-bound algorithms, i.e., by branching on the variables of one formulation but computing the relaxation of another. For example, Vance [Van98] tested branching rules induced by the assignment formulation while computing the model obtained by Dantzig-Wolfe decomposition of that. In the decomposed model, the relaxation is very strong but the symmetries remain, which is the reason why computational results were much worse than for branching on variables of the Gilmore-Gomory model. The same argument has prevented us from testing the subpattern models and/or branching rules induced by them. In contrast, the arc flow formulation seems to produce excellent branching rules [dC98, AdC03]. A new VRP-like formulation of 1D-BPP was proposed in [dAU03]. Thus, we think that it would be sensible to consider branching rules implied by the model, i.e., the branching on the number of certain subpatterns, but excluding symmetry.

# References

[AdC03] C. Alves and J. M. Valério de Carvalho, *A branch-and-price algorithm for integer variable sized bin-packing problems*, Tech. report, Universidade do Minho, Portugal, 2003.

[BS03] G. Belov and G. Scheithauer, *Setup and open stacks minimization in one-dimensional stock cutting*, Technical report, Dresden University, 2003.

[dAU03] M. Poggi de Aragão and Eduardo Uchoa, *Integer program reformulation for robust branch-abd-price algorithms*, Tech. report, Departamento de Informática, PUC-Rio, 2003.

[dC98] J. M. Valério de Carvalho, *Exact solution of cutting stock problems using column generation and branch-and-bound*, International Transactions in Operational Research **5** (1998), 35–44.

[dC02] _____, *LP models for bin-packing and cutting stock problems*, European Journal of Operational Research **141** (2002), no. 2, 253–273.

[Dyc81] H. Dyckhoff, *A new linear approach to the cutting stock problem*, Oper. Res. **29** (1981), no. 6, 1092–1104.

[GG61] P. C. Gilmore and R. E. Gomory, *A linear programming approach to the cutting-stock problem*, Operations Research **9** (1961), 849–859.

[Kan60] L. V. Kantorovich, *Mathematical methods of organizing and planning production*, Management Science **6** (1960), 363–422, (in Russian 1939).

[NST99] C. Nitsche, G. Scheithauer, and J. Terno, *Tighter relaxations for the cutting stock problem*, European Journal of Operational Research **112** (1999), 654–663.

[RS02] J. Rietz and G. Scheithauer, *Tighter bounds for the gap and non-IRUP constructions in the one-dimensional cutting stock problem*, Optimization **51** (2002), no. 6, 927–963.

[ST95] G. Scheithauer and J. Terno, *The modified integer round-up property of the one-dimensional cutting stock problem*, European Journal of Operational Research **84** (1995), 562–571.

[Sta88] H. Stadtler, *A comparison of two optimization procedures for 1– and 1,5–dimensional cutting stock problems*, OR Spektrum **10** (1988), 97–111.

[Van98] P. Vance, *Branch-and-price algorithms for the one-dimensional cutting stock problem*, Comput. Opt. Appl. **9** (1998), no. 3, 212–228.

[VBJN94] P. H. Vance, C. Barnhart, E. L. Johnson, and G. Nemhauser, *Solving binary cutting stock problems by column generation and branch-and-bound*, Comput. Opt. Appl. **3** (1994), 111–130.