

## Research Article

# The Effect of Welding on the One-Dimensional Cutting-Stock Problem: The Case of Fixed Firefighting Systems in the Construction Industry

Kolos Csaba Ágoston <sup>1,2</sup>

<sup>1</sup>Department of Operations Research and Actuarial Sciences, Corvinus University of Budapest, Fővám tér 13-15, 1093 Budapest, Hungary

<sup>2</sup>Institute of Economics, Research Centre for Economic and Regional Studies, Hungarian Academy of Sciences, Tóth Kálmán u. 4, 1097 Budapest, Hungary

Correspondence should be addressed to Kolos Csaba Ágoston; kolos.agoston@uni-corvinus.hu

Received 5 October 2018; Revised 11 January 2019; Accepted 31 January 2019; Published 3 March 2019

Academic Editor: Viliam Makis

Copyright © 2019 Kolos Csaba Ágoston. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Safety is paramount in the construction industry and the fixed sprinkler and water spray systems used in firefighting involve networks of pipes of various lengths. Manufacturers of such fixed firefighting systems need to either cut the existing stocks to length—a (one-dimensional) cutting-stock problem—or lengthen the existing stocks or leftover segments through welding, a (one-dimensional) cutting-stock problem with welding. Best industry practice safety requirements allow only one weld per length of pipe. The case of a Hungarian manufacturer of fixed firefighting systems motivates this article, which argues that the cutting-stock problem with welding (with single- or multiple-size stocks) may be converted to an equivalent cutting-stock problem (with multiple-size stocks). Readily available algorithms and software may then be used to generate an optimal cutting plan for the equivalent cutting-stock problem. Subject to certain restrictions, the optimal cutting plan for the equivalent cutting-stock problem may then be converted to cutting patterns for the original cutting-stock problem with welding.

## 1. Introduction

Safety is paramount in the construction industry and the fixed sprinkler and water spray systems used in firefighting are integral to a wide variety of buildings ranging from warehouses, through supermarkets, to educational establishments. Integrating fixed firefighting systems is in many cases mandatory, governed in the EU member countries by the European Standard EN 12845, currently incorporated, for example, in standards MSZ EN 12845:2015 in Hungary and BS EN 12845:2015 in the UK.

Fixed firefighting systems are assembled on site and involve networks of pipes (henceforth, *pipes*) of various lengths (henceforth, *sizes*) manufactured from the pipes in stock (henceforth, *stocks*). More often than not, manufacturers need to either cut the existing stocks to size (for example, from 6 meters to 4 meters from one stock, with a 2-meter leftover segment) or lengthen the existing stocks or leftover

segments through welding (for example, from 6 meters to 8 meters from two stocks, with a 4-meter leftover segment). Leftover segments may be welded to other stocks or to each other, but best industry practice safety requirements allow only one weld per pipe.

In the case of a Hungarian manufacturer of fixed firefighting systems, the company uses single-size, 6-meter stocks to produce pipes; Table 1 illustrates the sizes (14, in total) and quantities (31, in total) required for a particular network. Only one pipe size coincides with the stock size; 11 others are shorter and two are longer.

Manufacturing pipes for fixed firefighting systems is a one-dimensional cutting-stock problem with single- or multiple-size stocks and welding restricted to one weld per pipe (henceforth, *cutting-stock problem with welding*). This article sets out to find its optimal solution (that is, *optimal cutting-and-welding plan*) and argues that the readily available algorithms and software for the cutting-stock problem

TABLE 1: Illustrative pipe sizes and quantities for the case of a Hungarian manufacturer of fixed firefighting systems.

Stock size ( $S_1$ , in mm)	
6000	
Pipe sizes ( $s_i$ , in mm)	Pipe quantities ( $n_i$ )
2000	1
4000	1
4500	4
4660	2
4680	11
5000	1
5096	1
5250	1
5260	1
5500	1
5660	1
6000	2
7000	2
7200	2

with multiple-size stocks (henceforth, *cutting-stock problem*) may be used for the cutting-stock problem with welding. Numerical testing found the algorithm suggested in this article (see Section 5) adequate for the case of a Hungarian manufacturer of fixed firefighting systems.

Following this brief introduction, Section 2 reviews the relevant literature and positions this article in the ongoing debate fuelling research in the cutting-stock problem. Section 3 formulates the cutting-stock problem and the cutting-stock problem with welding as a mixed integer linear program, discusses cutting-and-welding pattern properties, and proves that optimal cutting-and-welding plans consist of minimal cutting-and-welding patterns. This characteristic of optimal cutting-and-welding plans is fundamental to the algorithm suggested in the next section. Section 4 discusses the practical difficulties caused by high cutting-and-welding stock quantities and nonsequential cutting-and-welding patterns. The ensuing need to limit the number of cutting-and-welding stocks and thus restrict the cutting-stock problem with welding shapes the algorithm suggested in this section. Generating minimal cutting-and-welding patterns and optimal cutting-and-welding plans pose possible difficulties, also explored in this section, alongside program running times, sequential patterns, and decision maker preferences. In Section 5 we apply the algorithm suggested in the previous section to the case of a Hungarian manufacturer of fixed firefighting systems with satisfactory results. Finally, Section 6 summarises the findings and suggests directions for further research.

## 2. Background

The cutting-stock problem—the question of how to optimise stock or how to cut stock to minimise cost—was first investigated by Kantorovich [1] and Kantorovich [2].

The column-generation algorithm suggested by Gilmore and Gomory [3] and Gilmore and Gomory [4] more than twenty years later became widely used by researchers and practitioners alike; Gilmore and Gomory [3] and Gilmore and Gomory [4] ‘were the first to present techniques practically applied to difficult real-world problems’ (Sweeney and Paternoster [5]). Complex cutting-stock problems generate complex cutting plans with very large numbers of feasible cutting patterns. Instead of attempting to generate and investigate them all at once, the Gilmore and Gomory algorithm first generates and investigates only a few. Subsequently generated cutting patterns replace existing ones if proven more efficient upon investigation, and through the iterative process the algorithm arrives at an optimal solution.

The strength of the Gilmore and Gomory algorithm lies in its ability to generate an optimal solution; its weakness in its inability to generate an optimal cutting plan where cutting pattern weights are integers. More often than not, rounding decimal cutting pattern weights to the nearest integer results in infeasible or suboptimal cutting plans and/or overproducing some orders and underproducing others. Generating an optimal cutting plan for the cutting-stock problem thus became the focus of further research; Roodman [6], Wäscher and Gau [7], and Gradisar and Trkman [8] suggested heuristic methods, while Dyckhoff [9], Scheithauer and Terno [10], Vance et al. [11], Vanderbeck [12], and Belov and Scheithauer [13] proposed further algorithms. Sweeney and Paternoster [5] and Wäscher et al. [14] provided comprehensive literature reviews of these and other—usually heuristic—methodologies.

The cutting-stock problem therefore stayed in operations research focus even in its simplest, original, one-dimensional definition; in recent years, for example, Levine and Ducatelle [15] and Araújo et al. [16] used evolutionary algorithm for the cutting-stock problem. Particularly relevant to this article though are the works of Cui and Yang [17], who investigated leftover segment recycling.

Cutting stock is a nondeterministic polynomial-time-(NP-) hard problem (Cheng et al. [18]), which may require exponential software running times; nonetheless, in practice, even large samples may be investigated for optimal solutions (see Goulimis [19]; Belov and Scheithauer [13]). Therefore this article assumes that there are algorithms and software for obtaining the exact (global) solution of a cutting-stock problem as defined in Section 1 (see Wäscher et al. [14] for typology, including MSSCSP). If the goal is to design business applications where suboptimal solutions are acceptable, then exact (global) solutions are not necessary, because good (heuristic) solutions are sufficient.

Zak [20] defines the skiving stock problem, a counterpart to the cutting-stock problem, as “find the maximum number of items with minimum length  $L$  that can be constructed by connecting a given supply of smaller item lengths” (Martynovic and Scheithauer [21]).

In the skiving stock problem, the number of connected items is not limited. In our case, it is crucial that every item can be constructed from two pieces at most. The investigated cutting-and-welding problem is a combination of the cutting-stock problem and the skiving stock problem. Tanir et al.

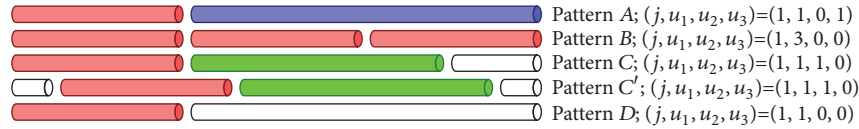


FIGURE 1: Some feasible patterns for cutting 2-, 3-, and 4-meter pipes from single-size, 6-meter stocks.

[22] examined the possibility of welding in a cutting-stock problem and suggested a heuristic method. There were no restrictions on the number of welds per pipe, while in this paper the number of welds per pipe is restricted to one.

### 3. Theory

**3.1. Formulating the Cutting-Stock Problem.** The cutting-stock problem consists of generating the optimal—minimal cost—cutting plan.  $I$  pipes of various sizes  $s_i$  ( $i = 1, 2, \dots, I$ ) need to be cut in various quantities  $n_i$  from  $J$  stocks of various sizes  $S_j$  ( $S_1, S_2, \dots, S_J$ ). Feasible cutting patterns may be represented by sets of integers  $(j, u_1, u_2, \dots, u_I)$ , where  $j$  is the stock index and  $u_i$  is the number of pipes of size  $s_i$  produced from stock  $S_j$ . Evidently,  $\sum_{i=1}^I u_i s_i \leq S_j$  and  $\sum_{i=1}^I u_i \geq 1$ .

For example,  $s_1 = 2$ -,  $s_2 = 3$ -, and  $s_3 = 4$ -meter pipes could be cut from single-size,  $S_1 = 6$ -meter stocks in various feasible patterns  $(j, u_1, u_2, u_3)$ . Figure 1 illustrates five such cutting patterns: the 6-meter stocks are cut into one 2-meter pipe and one 4-meter pipe ( $6 = 2 + 4$ ), in pattern A (1, 1, 0, 1); into three 2-meter pipes ( $6 = 2 + 2 + 2$ ), in pattern B (1, 3, 0, 0); into one 2-meter pipe, one 3-meter pipe, and one 1-meter leftover segment ( $6 = 2 + 3 + 1$ ), in pattern C (1, 1, 1, 0); into one 2-meter pipe, one 3-meter pipe, and two 0.5-meter leftover segments ( $6 = 0.5 + 2 + 3 + 0.5$ ), in pattern C' (1, 1, 1, 0); and into one 2-meter pipe and one 4-meter leftover segment ( $6 = 2 + 4$ ), in pattern D (1, 1, 0, 0). Cutting patterns reflect what is produced, not how is produced; as such, patterns C and C' are equivalent and pattern D is wasteful.

If  $\mathcal{P}$  represents all the feasible cutting patterns, then cutting pattern  $p_\ell \in \mathcal{P}$  produces  $u_{i\ell}$  pipes of size  $s_i$  at the pattern price  $\pi_\ell$ , the stock price, that is, if the cutting cost is ignored as insignificantly small.

Thus the cutting-stock problem may be formulated as the following mixed integer linear program:

$$\begin{aligned} \min_b \quad & \sum_{\ell: p_\ell \in \mathcal{P}} b_\ell \pi_\ell \\ \text{s.t.} \quad & \sum_{\ell: p_\ell \in \mathcal{P}} b_\ell u_{i\ell} \geq n_i \quad \forall i \\ & b_\ell \in \mathcal{N}, \end{aligned} \tag{1}$$

where every feasible solution represents a cutting plan, an optimal cutting plan is a feasible solution with minimal cost, and the decision variable  $b_\ell \in \mathcal{N}$  indicates how many times a cutting plan uses cutting pattern  $p_\ell$ .

Due to its exponential size, solving the mixed integer linear program is not practical for large instances. In practice, this should be done with a combination of branch-and-bound

and column-generation (see Belov and Scheithauer [13] for branch-and-price and column-generation).

Optimal cutting plans do not exclude wasteful cutting patterns like pattern D in Figure 1; and a cutting-stock problem may have more than one optimal cutting plan. For example, if four 2-meter pipes need to be cut from single-size, 6-meter stocks, then one optimal cutting plan could use two patterns to cut one 6-meter stock in three 2-meter pipes ( $6 = 3 \times 2$ ) and another 6-meter stock in one 2-meter pipe and one 4-meter leftover segment ( $6 = 2 + 4$ ). Another optimal plan could use the first pattern twice to cut two 6-meter stocks in six 2-meter pipes ( $2 \times 6 = 6 \times 2$ ), two 2-meter pipes too many. Subject to additional considerations such as number of cuts, amount of waste and size of overproduction, decision makers may prefer one optimal cutting plan to another.

**3.2. Formulating the Cutting-Stock Problem with Welding.** Similarly to the cutting-stock problem, the cutting-stock problem with welding consists of generating the optimal—minimal cost—cutting-and-welding plan. The number of feasible cutting-and-welding patterns is just larger than the number of feasible cutting patterns. For example, if 4-meter pipes need to be produced from single-size, 6-meter stocks, then only one cutting pattern is feasible in the cutting-stock problem as opposed to two cutting-and-welding patterns feasible in the cutting-stock problem with welding. In the cutting-stock problem, the 6-meter stocks can only be cut into 4-meter pipes and wasteful 2-meter leftover segments. In the cutting-stock problem with welding, the ‘wasteful’ 2-meter leftover segments may be welded together, in additional 4-meter pipes (see patterns A and B in Figure 2). However, whether pattern B is preferable to pattern A depends on the price per weld relative to the price per stock, if higher, then pattern A prevails.

In the cutting-stock problem with welding, feasible cutting-and-welding patterns may be represented by sets of integers  $(\mathbf{U}, \mathbf{u}) = (U_1, U_2, \dots, U_J, u_1, u_2, \dots, u_I)$  and unique assignment matrices  $A$  with  $\sum_{j=1}^J U_j$  rows,  $\sum_{i=1}^I u_i$  columns, and only zero and one as elements.  $U_j$  represents the number of stocks of size  $S_j$ ,  $j = 1, \dots, J$ , used to produce  $u_i$  pipes of size  $s_i$ ,  $i = 1, \dots, I$ , through cutting and welding. Thus, the first row represents the first stock of size  $S_1$ , the second row the second stock of size  $S_1$ , ..., and the  $U_1$ th row the  $U_1$ th stock of size  $S_1$ . Similarly, the  $(U_1 + 1)$ th row represents the first stock of size  $S_2$ , the  $(U_1 + 2)$ th row the second stock of size  $S_2$ , ..., the  $(U_1 + U_2)$ th row the  $U_2$ th stock of size  $S_2$ , and so on. Also, the first column represents the first pipe of size  $s_1$ , the second column the second pipe of size  $s_1$ , ..., and the  $u_1$ th column the  $u_1$ th pipe of size  $s_1$ . Similarly, the  $(u_1 + 1)$ th column represents the first pipe of size  $s_2$ , the  $(u_1 + 2)$ th column the second pipe of

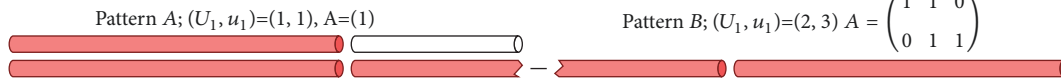


FIGURE 2: Patterns for producing 4-meter pipes from single-size, 6-meter stocks through cutting and cutting and welding.

size  $s_2, \dots$ , the  $(u_1 + u_2)$ th column the  $u_2$ th pipe of size  $s_2$ , and so on. If rows are indexed by  $r$ , columns by  $c$ , and elements by  $a_{rc}$ , then  $a_{rc} = 1$  implies that at least some segment of pipe  $c$  is produced with at least some segment of stock  $r$ . This article assumes that no stock size and no stock is redundant and that therefore  $a_{rc} = 1$  at least once in every row and once in every column. Since any one stock may be assigned to any number of pipes, but any one pipe may only be produced from two stocks at the most,  $a_{rc} = 1$  no more than twice in every column.

The set of integers  $(\mathbf{U}, \mathbf{u})$  and assignment matrix  $A$  represent a feasible cutting-and-welding pattern if an equal-size welding matrix  $W$  can be found to offer a feasible welding option. If rows are indexed by  $r$ , columns by  $c$ , and elements by  $w_{rc}$ , then  $w_{rc} > 0$  if and only if  $a_{rc} = 1$ . Also, if  $S(r)$  is the size of the stock represented by row  $r$  and  $s(c)$  is the size of the pipe represented by column  $c$ , then the sum of all the elements in any one row is less than  $S(r)$  and the sum of all the elements in any one column is equal to  $s(c)$ . Furthermore,  $\sum_{j=1}^J U_j \geq 1$  and  $\sum_{i=1}^I u_i \geq 1$ . In addition, one has the following:

- (i) A pipe size shorter than the sum of all the stock sizes is insufficient in itself. For example, if  $S_1 = 1$  meter,  $S_2 = 6$  meters, and  $s_1 = 4$  meters, then  $(U_1, U_2, u_1) = (4, 0, 1)$  requires three welds and is not a valid cutting-and-welding pattern.
- (ii) Cutting-and-welding patterns reflect what is produced, not how is produced. For example, from a cutting-stock problem with welding point of view, patterns  $C$  and  $C'$  in Figure 1 have identical sets of integers  $(U_1, u_1, u_2, u_3) = (1, 1, 0, 1)$  and assignment matrices  $A = (1 \ 1)$  and welding matrices  $W = (2000 \ 3000)$  (see Section 3.1).

For another example, 3.6-meter pipes can be produced from single-size, 6-meter stocks in various cutting-and-welding patterns, for example, by cutting one stock in one 3.6-meter pipe and one 2.4-meter segment ( $6 = 3.6 + 2.4$ ), by cutting another stock in one 3.6-meter pipe and two 1.2-meter segments ( $6 = 1.2 + 3.6 + 1.2$ ), by welding the 2.4-meter segment with one of the two 1.2-meter segments ( $2.4 + 1.2 = 3.6$ ), with  $(U_1, u_1) = (2, 3)$ ,  $A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ , and  $W = \begin{pmatrix} 3600 & 2400 & 0 \\ 0 & 1200 & 3600 \end{pmatrix}$ , by cutting each of two stocks in one 3.6-meter pipe, one 1.8-meter segment, and one 0.6-meter leftover segment ( $2 \times 6 = 2 \times 3.6 + 2 \times 1.8 + 2 \times 0.6$ ), and by welding the two 1.8-meter segments ( $1.8 + 1.8 = 3.6$ ), with identical  $(U_1, u_1) = (2, 3)$  and  $A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$  as before, but different  $W = \begin{pmatrix} 3600 & 1800 & 0 \\ 0 & 1800 & 3600 \end{pmatrix}$ , and so on.

- (iii) Redundancies can appear in cutting-and-welding patterns, but these do not affect optimal solutions.

For example, by cutting each of two 6-meter stocks in one 3.6-meter pipe, one 1.8-meter segment, and one 0.6-meter leftover segment ( $2 \times 6 = 2 \times 3.6 + 2 \times 1.8 + 2 \times 0.6$ ), one in three 3.6-meter pipes is welded. Six identical sets of integers— $(U_1, u_i) = (2, 3)$ —and six different assignment matrices— $A_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ ,  $A_2 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ ,  $A_3 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ ,  $A_4 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ ,  $A_5 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$ , and  $A_6 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ —correspond to six different cutting-and-welding patterns with identical practical outcomes.

- (iv) Welding is not a compulsory component of cutting-and-welding patterns. For example, two 5-meter pipes can be produced from two 6-meter stocks by cutting each of the two 6-meter stocks in one 5-meter pipe and one 1-meter leftover segment ( $2 \times 6 = 2 \times 5 + 2 \times 1$ ), with  $(U_1, u_1) = (2, 2)$  and  $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  as if the cutting patterns  $(U_1, u_1) = (1, 1)$  and  $A = (1)$  were used twice.
- (v) The number of cutting-and-welding patterns is finite, since the number of pipes and the number of stocks are both finite.
- (vi) Any one pipe can only be produced from two stocks at the most, but a cutting-and-welding pattern may involve more than two stocks, which may be cut and welded in various ways.

For example, six 5-meter pipes can be produced from five 6-meter stocks by cutting each of two 6-meter stocks in one 5-meter pipe and one 1-meter segment ( $2 \times 6 = 2 \times 5 + 2 \times 1$ ), by cutting each of two 6-meter stocks in one 4-meter segment and one 2-meter segment ( $2 \times 6 = 2 \times 4 + 2 \times 2$ ), by cutting one 6-meter stock in two 3-meter segments ( $6 = 2 \times 3$ ), by welding each of the two 1-meter segments to one of the two 4-meter segments ( $2 \times 1 + 2 \times 4 = 2 \times 5$ ), and by welding each of the two 2-meter segments to one of the two 3-meter segments ( $2 \times 2 + 2 \times 3 = 2 \times 5$ ) (see Figure 3). This cutting-and-welding pattern is therefore represented by  $(U_1, u_1) = (5, 6)$ ,

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

$$W = \begin{pmatrix} 5000 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 4000 & 2000 & 0 & 0 & 0 \\ 0 & 0 & 3000 & 3000 & 0 & 0 \\ 0 & 0 & 0 & 2000 & 4000 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 5000 \end{pmatrix} \quad (2)$$

If  $\mathcal{Q}$  represents all the feasible cutting-and-welding patterns, then cutting-and-welding pattern  $q_\ell$  produces  $u_{i\ell}$  pipes of size  $s_i$  at the pattern price  $\pi_\ell$ , the stock and welding prices, that is, if the cutting cost is ignored as insignificantly small.

Similar to the cutting-stock problem, the cutting-stock problem with welding may be formulated as the following mixed integer linear program:

$$\begin{aligned} \min_{\mathbf{b}} \quad & \sum_{\ell: q_\ell \in \mathcal{Q}} b_\ell \pi_\ell \\ \text{s.t.} \quad & \sum_{\ell: q_\ell \in \mathcal{Q}} b_\ell u_{i\ell} \geq n_i \quad \forall i \\ & b_\ell \in \mathcal{N}, \end{aligned} \quad (3)$$

where every feasible solution represents a cutting-and-welding plan, an optimal cutting-and-welding plan is a feasible solution with minimal cost, and the decision variable  $b_\ell \in \mathcal{N}$  indicates how many times a cutting-and-welding plan uses cutting-and-welding pattern  $q_\ell$ .

Similar to the cutting-stock problem, due to its exponential size, this method is not practical for solving large instances. In practice, this should be replaced with a combination of branch-and-bound and column-generation techniques (see Belov and Scheithauer [13], for example, for branch-and-price and column-generation techniques).

### 3.3. The Relation between Optimal Cutting-and-Welding Plans and Minimal Cutting-and-Welding Patterns

**Proposition 1.** *Every optimal cutting-and-welding plan can be constructed using cutting-and-welding patterns where the number of welds involved is one fewer than the number of stocks used. We call these cutting-and-welding patterns minimal.*

*Proof.* Let us prove this proposition by contradiction and assume that there exists an optimal cutting-and-welding plan with a cutting-and-welding pattern with  $k$  stocks (rows  $r_1, r_2, \dots, r_k$  in the assignment matrix  $A$ ) and  $w$  welds, where  $w \geq k$ . This cutting-and-welding pattern may be represented as a graph—a network of nodes and linking edges, where the nodes represent the stocks and the edges represent the welds—where there may be no edge, one edge, or two or more edges between any two nodes. If  $w \geq k$ , then the network is cyclical; let  $r_1, r_2, \dots, r_m$  be the stocks that form the cycle  $r_1 \rightarrow r_2 \rightarrow \dots \rightarrow r_m \rightarrow r_1$ . Corresponding indices (columns  $c_1, c_2, \dots, c_m$  in the assignment matrix  $A$ ) need to be found so that all the cyclical matrix elements  $a_{r_1 c_1}, a_{r_1 c_2}, a_{r_2 c_2}, a_{r_2 c_3}, a_{r_3 c_3}, a_{r_3 c_4}, \dots, a_{r_{m-1} c_m}, a_{r_m c_m}$ , and  $a_{r_m c_1}$  are one. If  $\varepsilon$

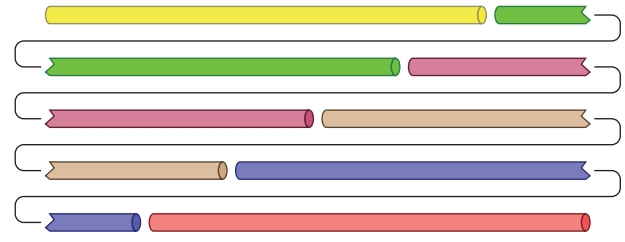


FIGURE 3: Patterns for cutting-and-welding six 5-meter pipes from five 6-meter stocks.

is the smallest among  $w_{r_1 c_1}, w_{r_2 c_2}, \dots, w_{r_m c_m}$  elements in the corresponding welding matrix  $W$ , then subtract  $\varepsilon$  from all of these elements and adding  $\varepsilon$  to all the elements  $w_{r_1 c_2}, w_{r_2 c_3}, \dots, w_{r_{m-1} c_m}$ , and  $w_{r_m c_1}$  results, among others, in a zero element, in other words, a pipe without welding and a cutting-and-welding pattern with one fewer welds. Modifications in  $W$  lead of course to modifications in  $A$ ; if  $w_{rc} > 0$ , then  $a_{rc} = 1$ ; if  $w_{rc} = 0$ , then  $a_{rc} = 0$ ; and if  $w - 1 \geq k$ , then the process requires as many iterations as necessary to obtain a cycle-free graph, a tree or a forest. A contradiction is that since a tree where the number of welds involved is one fewer than the number of stocks used represents a minimal cutting-and-welding pattern.  $\square$

This characteristic of optimal cutting-and-welding plans plays a fundamental role in the algorithm suggested in this article (see Section 4.1).

Important to note that Proposition 1 does not state that welds are always necessary. If we avoid welds, the number of used stock is 1, and the number of welds is 0, so Proposition 1 holds.

## 4. Calculation

By adding their lengths,  $k$  single- or multiple-size cutting-and-welding stocks may be converted to one equivalent cutting-stock and the cutting-stock problem with welding may be converted to an equivalent cutting-stock problem. If  $\pi_w$  is the price per weld, then the price of the equivalent cutting stock is the price of the  $k$  cutting-and-welding stocks plus  $(k - 1)\pi_w$ . Multiple-size cutting-and-welding stocks (of  $J$  sizes) may generate large numbers of equivalent cutting stocks, given by the combination  $\binom{k+J-1}{k}$ . If ‘+’ represents one stock and ‘/’ the border between two stock sizes, then, for example, ‘+++/++++’ represents two stocks from the first size, one from the second, none from the third, and three from the fourth. For  $k$  cutting-and-welding stocks and  $J$  cutting-and-welding stock sizes, there are  $J - 1$  ‘/’ signs and  $k + J - 1$  places where the  $k$  ‘+’ signs could go. In the example above,  $k = 6, J = 4$ , and  $\binom{9}{6} = 84$ .

Cutting-and-welding patterns are not necessarily sequential; for example, the first stock could be cut in two segments, one to be welded with a segment from the fourth stock and the other with a segment from the tenth stock. High cutting-and-welding stock quantities and nonsequential cutting-and-welding patterns cause practical difficulties that may be

reduced by reducing the cutting-and-welding stock quantities.

Let us therefore limit  $k$  to  $\kappa$ , and thus bound the cutting-stock problem with welding; the smaller the  $\kappa$ , the smaller the number of equivalent cutting patterns. For single-size cutting-and-welding stocks (as in the case which motivates this article), the equivalent cutting-stock problem tolerates quite large sizes of  $\kappa$  (see Section 4.1). Generally, however, finding the optimal, smallest  $\kappa$  is of obvious importance (see Section 4.3).

**4.1. An Algorithm for Solving the Bounded Cutting-Stock Problem with Welding.** To solve the bounded cutting-stock problem with welding, this article suggests a three-step algorithm that—at the least—is a good heuristic and that—at the best—generates an optimal cutting-and-welding plan:

- (1) Convert the bounded cutting-stock problem with welding to an *equivalent cutting-stock problem* as follows:
  - (a) Choose  $k \leq \kappa$  single- or multiple-size *bounded cutting-and-welding stocks* in all the ways possible; for example, 5- and 6-meter cutting-and-welding stocks and  $\kappa = 3$  generate two bounded cutting-and-welding stocks for  $k = 1$  (5 and 6 meters); three for  $k = 2$  (5 + 5 = 10, 5 + 6 = 11, and 6 + 6 = 12 meters); and four for  $k = 3$  (5 + 5 + 5 = 15, 5 + 5 + 6 = 16, 5 + 6 + 6 = 17, and 6 + 6 + 6 = 18 meters).
  - (b) Add their lengths, to convert the bounded cutting-and-welding stocks to one *equivalent cutting stock*.
  - (c) Add their prices to the  $(k - 1)\pi_w$  welding price, to obtain the equivalent cutting-stock price.
- (2) Solve the equivalent cutting-stock problem (that is, generate the optimal equivalent cutting plan) with the aid of open-source or commercial software (see Section 3.1).
- (3) Convert each equivalent cutting pattern to a *bounded cutting-and-welding pattern* (henceforth, *pattern conversion problem*, see Section 4.2).

This algorithm poses two possible difficulties.

First (see step (1)), multiple-size bounded cutting-and-welding stocks may generate alternative equivalent cutting stocks. For example, if the multiple-size bounded cutting-and-welding stocks consist of 4- and 6-meter stocks, then three 4-meter stocks ( $3 \times 4 = 12$ ) with two welds and two 6-meter stocks ( $2 \times 6 = 12$ ) with one weld generate two alternative equivalent cutting stocks of identical, 12-meter sizes, but nonidentical prices. To solve the equivalent cutting-stock problem (see step (2)), good practice suggests that the algorithm should consider the cheapest equivalent cutting stock.

Second (see step (3)), to generate an (optimal) bounded cutting-and-welding plan, the optimal equivalent cutting plan would need to be converted to minimal bounded

cutting-and-welding patterns (see Section 4.2). This difficulty is trivial in most cases, but by no means in all. For example, if 10-meter pipes need to be produced from single-size, 6-meter cutting-and-welding stocks, then five 6-meter stocks and four welds would generate a 30-meter equivalent cutting stock and two cuts would generate three 10-meter pipes ( $5 \times 6 = 30 = 3 \times 10$ ). However, since one of the pipes would have two welds instead of the maximum one, this cutting-and-welding pattern is not even feasible let alone minimal. The algorithm would not only need to solve the equivalent cutting-stock problem (see Section 3.1); it would also need to tackle extreme cases, with no minimal bounded cutting-and-welding patterns (see Section 4.2).

**4.2. Formulating the Pattern Conversion Problem.** Let us consider one of the equivalent cutting patterns with  $K$  single- or multiple-size stocks and  $L$  single- or multiple-size pipes— $k = 1, 2, \dots, K$  and  $\ell = 1, 2, \dots, L$ —generated at step (2) (see Section 4.1). Also, let  $S_k$  be the size of stock  $k$ ,  $x_{\ell k}$  the decision variable which shows the size of segment of pipe  $\ell$  produced from stock  $k$ , and  $B_{\ell k} \in \{0; 1\}$  the binary variable which shows whether  $x_{\ell k}$  is zero or positive.

Thus the pattern conversion problem may be formulated as the following mixed integer linear program:

$$\begin{aligned}
 & \sum_{\ell=1}^L \sum_{k=1}^K B_{\ell k} \longrightarrow \min \\
 \text{s.t.: } & x_{\ell k} - S_k B_{\ell k} \leq 0 \quad \ell = 1 \dots L, \quad k = 1 \dots K \\
 & \sum_{k=1}^K x_{\ell k} = s_{\ell} \quad \ell = 1 \dots L \\
 & \sum_{\ell=1}^L x_{\ell k} \leq S_k \quad k = 1 \dots K \\
 & \sum_{k=1}^K B_{\ell k} \leq 2 \quad \ell = 1 \dots L \\
 & x_{\ell k} \geq 0 \quad \ell = 1 \dots L, \quad k = 1 \dots K \\
 & B_{\ell k} \in \{0; 1\} \quad \ell = 1 \dots L, \quad k = 1 \dots K
 \end{aligned} \tag{4}$$

Converting equivalent cutting patterns to minimal bounded cutting-and-welding patterns may seem trivial and in most cases indeed it is. However, multiple-size stocks and long pipes which all need to be welded complicate conversion and an equivalent cutting pattern might not convert to a minimal bounded cutting-and-welding pattern.

The value of the objective function indicates the number of stocks and/or stock segments necessary to produce the  $L$  pipes. Each pipe contains at least one stock or stock segment, but not more than two; thus, the value of the objective function is given by the number of pipes and the number of welds:

- (i) Thus if the value is  $L + K - 1$ , then the bounded cutting-and-welding pattern generated is minimal (see Section 3.3).

- (ii) If the value is less than  $L + K - 1$ , then the software used at step (2) (see Section 4.1) was inadequate, the equivalent cutting plan was suboptimal, and the bounded cutting-and-welding pattern generated is feasible, but not minimal. In practice, albeit with no firm claim to optimality, this feasible pattern may be broken down into smaller, minimal patterns; after all, every pipe can be produced from only two stocks at the most; tree graphs such as those discussed in Section 3.3 are noncyclical even for number of welds smaller than  $K - 1$ . Since the number of stocks is identical, but the number of welds is smaller, this set of minimal patterns is cheaper than the feasible pattern generated by the pattern conversion problem. However, further theoretical investigation is outside the scope of this article.
- (iii) If the pattern conversion problem does not have a feasible solution, then the software used at step (2) (see Section 4.1) is not necessarily inadequate. In practice, albeit with no firm claim to optimality, a feasible bounded cutting-and-welding pattern may be obtained through iteration by increasing the number of stocks in the pattern conversion problem from  $K$  to  $K + 1$  and  $K + 2$ , and so on. In the case of multiple-size stocks, increasing the number of longest-size stocks will definitely result in a feasible solution; in certain cases, increasing the number of shorter-size stocks may result in a feasible solution too. As before, this feasible pattern with  $K'$  stocks and fewer than  $K' - 1$  welds may be broken down into smaller, minimal patterns.

#### 4.3. The Relation between Optimal Equivalent Cutting Plans, Optimal Bounded Cutting-and-Welding Plans, and Optimal Cutting-and-Welding Plans

**Proposition 2.** *If every equivalent cutting pattern of the optimal equivalent cutting plan can be converted to a minimal bounded cutting-and-welding pattern, then the algorithm generates an optimal solution for the bounded cutting-stock problem with welding.*

*Proof.* Let us prove this proposition by contradiction and assume that the algorithm does not generate an optimal solution for the bounded cutting-stock problem with welding, although every equivalent cutting pattern of the optimal equivalent cutting plan can be converted to a minimal bounded cutting-and-welding pattern. Then there exists a bounded cutting-and-welding plan cheaper than the bounded cutting-and-welding plan generated by the algorithm. Since all the bounded cutting-and-welding patterns of this bounded cutting-and-welding plan can be converted to equivalent cutting patterns, then the equivalent cutting plan generated is cheaper than the optimal equivalent cutting plan generated by the algorithm; that is a contradiction.  $\square$

**Proposition 3.** *The optimal solution for the bounded cutting-stock problem with welding is an optimal solution for the cutting-stock problem with welding too if  $\kappa$  is sufficiently large.*

*Proof.* The algorithm suggested in this article for solving cutting-stock problems with welding relies on limiting the available cutting-and-welding stocks to  $\kappa$  by bounding the cutting-stock problem with welding (see Section 4.1). Since every pipe can be produced from two stocks at the most,  $\kappa$  has an upper limit of  $2 \sum_{i=1}^I n_i$ .  $\square$

However, this upper bound may still be too large for practical purposes and finding the lowest upper bound is important (the lower the upper bound, the fewer the number of equivalent cutting stocks), but problematic even for single-size cutting-and-welding stocks. Moreover, even the lowest upper bound may be too large for practical purposes. For example, let us consider producing  $n_1$  1,001-meter pipes from single-size, 1-meter cutting-and-welding stocks. If  $n_1 = 1,000$ , then  $\kappa \geq 2 \sum_{i=1}^I n_i = 2 \times n_1 = 2 \times 1,000 = 2,000$ , but the lowest upper bound is evidently 1,001, if the cutting cost is ignored as insignificantly small. Finding the lowest upper bound is therefore possible, in special cases, but generally problematic, particularly for multiple-size cutting-and-welding stocks.

**4.4. Algorithm Running Times.** The algorithm running time consists of the running time for the equivalent cutting-stock problem (see step (2), Section 4.1) and the running time for the pattern conversion problem (see step (3), Section 4.1).

To investigate running times for the equivalent cutting-stock problem, the BarCut (version 2.1) program was run on a computer with an Intel Duo 2.33 GHz processor, 6 GB of random access memory (RAM), and Windows 7 Enterprise as operating system. At less than 9 seconds, the running times for the case which motivates this article proved acceptable (see Section 5).

To investigate running times for the pattern conversion problem, the GLPK (version 4.55) program was run on the same computer (see Table 2, where the second column shows the equivalent cutting patterns—generated by a complex, hypothetical solution of the equivalent cutting-stock problem—that need to be converted to feasible cutting-and-welding patterns). If the pattern conversion problem did not have an optimal solution, the running times proved short; if it did, the running times proved very long at times and in obvious need of further analysis.

Moderate numbers of pipes (see the second column in Table 2) and moderate numbers of single-size, 6-meter cutting-and-welding stocks (see the third column in Table 2) generate feasible solutions in (milli)seconds (see column G1 in Table 2). Large numbers may require long running times, but feasible solutions can generally be generated in (milli)seconds and the actual value of the objective function ( $\sum_{\ell=1}^L \sum_{k=1}^K B_{\ell k}$ ) indicates the existence of a feasible solution (see Section 4.2). The ‘normal’ optimal value of the objective function is  $L + K - 1$  (see Section 4.2); however, more efficient cutting-and-welding patterns may be generated, with fewer

TABLE 2: Algorithm running times for minimal cutting-and-welding patterns (\* = no minimal solution).

Pattern	Pipes ( $s_\ell$ , in mm)	Stocks ( $K$ )	G1	G2	G3	G4	G5	S1
A	2070, 2140, 2210, 2280, 2350, 2420, 4530	3	0.4	0.1	0.1	0.1	0.1	0.1
B	2070, 2140, 2210, 2280, 2350, 2420, 2490, 2560, 2630, 2850	4	> 3600	0.1	0.1	0.1	0.1	0.1
C	2070, 2140, 2210, 2280, 2350, 2420, 2490, 2560, 2630, 2700, 2770, 3380	5	> 3600	0.1	0.1	0.1	0.1	0.1
D	2070, 2140, 2210, 2280, 2350, 2420, 2490, 2560, 2630, 2700, 2770, 2840, 2910, 3630	6	> 3600	0.1	0.1	0.1	> 3600	0.1
E	2070, 2140, 2210, 2280, 2350, 2420, 2490, 2560, 2630, 2700, 2770, 2840, 2910, 2980, 3050, 3600	7	> 3600	0.1	0.1	0.1	> 3600	0.2
F	2070, 2140, 2210, 2280, 2350, 2420, 2490, 2560, 2630, 2700, 2770, 2840, 2910, 2980, 3050, 3120, 3190, 3290	8	> 3600	0.4	0.6	1.7	> 3600	0.2
G	2070, 2140, 2210, 2280, 2350, 2420, 2490, 2560, 2630, 2700, 2770, 2840, 2910, 2980, 3050, 3120, 3190, 3260	8	> 3600	-	-	-	-	-
H	10000, 10000, 10000	5	0.1*	-	-	-	-	-
I	11000, 11000, 11000, 3000, 3000, 3000	7	86.6	0.0	0.0	1.0*	-	0.1*
J	1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 11001, 11001, 11001, 11001, 11001, 11001, 11001, 11001, 11493, 11499	20	> 3600	118.4	> 3600	> 3600	> 3600	> 3600

than  $K - 1$  welds (see pattern G in Table 2, with  $K - 2 = 8 - 2 = 6$  welds). (Pattern H in Table 2 has no minimal solution.)

Even with  $\sum_{\ell=1}^L \sum_{k=1}^K B_{\ell k} = L + K - 1$  as constraints, where appropriate (see Section 4.4.1), the algorithm running time for the optimal solution is generally a matter of (milli)seconds (see column G2 in Table 2).

**4.4.1. Decision Maker Preferences and Sequential Cutting-and-Welding Patterns.** Equivalent cutting patterns may convert to alternative cutting-and-welding patterns and decision makers may prefer some to others. Preferences may be easily incorporated by changing the objective function from  $\sum_{\ell=1}^L \sum_{k=1}^K B_{\ell k}$  (see Section 4.2) to  $\sum_{\ell=1}^L o_\ell (\sum_{k=1}^K B_{\ell k})$ , where  $o_\ell$  represents decision maker's score point (that is, preference level) for pipe  $\ell$ ; lower  $o_\ell$  values indicate higher preferences for welding. Since longer pipes are likely to be welded from stocks, not just cut, we may logically attribute an  $o_1 = 0$  to the longest pipe, an  $o_2 = 1$  to the second-longest pipe, ..., and an  $o_L = L - 1$  to the shortest pipe. Without the  $\sum_{\ell=1}^L \sum_{k=1}^K B_{\ell k} = L + K - 1$  constraints, the program running times for the pattern conversion problem might be long and an alternative, two-step method preferable. In step one, the program is allowed to run for the few (milli) seconds necessary to calculate the minimal number of welds; arriving at the optimal value of the objective function is a fast process; investigating all the possible alternatives is not.

In step two, the objective function is changed from  $\sum_{\ell=1}^L \sum_{k=1}^K B_{\ell k}$  to  $\sum_{\ell=1}^L o_\ell (\sum_{k=1}^K B_{\ell k})$  and the  $\sum_{\ell=1}^L \sum_{k=1}^K B_{\ell k} = L + K - 1$  constraints or alternative constraints, based on step one, are added to generate a preferred pattern hopefully quickly (see column G3 in Table 2). Pattern J in Table 2

represents an exceptionally difficult case, where a minimal pattern can be generated relatively quickly only if the decision maker's preferences are disregarded.

Understandably, decision makers prefer sequential cutting-and-welding patterns, where pipes are placed in relevant order and the end of one stock is welded to the beginning of the next. In the tree graph discussed in Section 3.3, sequential patterns would correspond to paths.

When all the pipes are shorter than the shortest stock, sequential patterns are evidently possible; in fact, every permutation yields a sequential pattern. When some of the pipes are longer than the shortest stock, sequential patterns are not always possible. For example, if 4-, 5-, and 9-meter pipes need to be produced from single-size, 6-meter stocks, then the permutation (9, 5, 4) generates a sequential pattern while the permutation (5, 9, 4) does not, because the 9-meter pipe would require two welds. For another example, there is no sequential pattern if three 11-meter and three 3-meter pipes need to be produced from single-size, 6-meter stocks, but there is a minimal, nonsequential pattern involving seven stocks, six welds, and five cuts (see Figure 4). (Six stocks, three welds, and three cuts will generate three 11-meter pipes and three 1-meter segments. The seventh stock will be cut in three 2-meter segments, each to be welded to one of the three 1-meter segments to generate three 3-meter pipes.)

Sequential cutting-and-welding patterns may be generated by adding the constraints

$$B_{k\ell} + B_{k_j} \leq 1, \quad (5)$$

$$\ell = 1 \dots L - 3, \quad j = \ell + 2 \dots L, \quad k = 1 \dots K$$



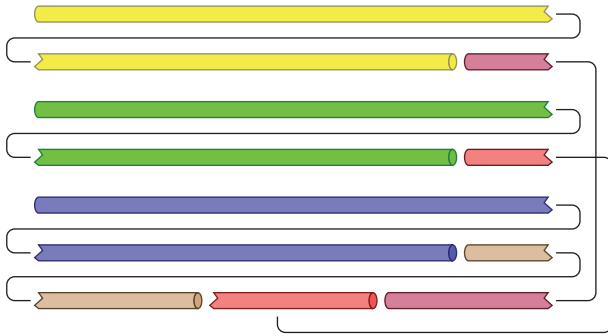


FIGURE 4: Nonsequential cutting-and-welding pattern for producing three 11-meter pipes and three 3-meter pipes from seven single-size, 6-meter stocks.

and  $\sum_{\ell=1}^L \sum_{k=1}^K B_{\ell k} = L + K - 1$  to the objective function  $\sum_{\ell=1}^L o_{\ell} (\sum_{k=1}^K B_{\ell k})$  (see Section 4.2).

The number of additional constraints increases quadratically in the number of stocks. If decision maker preferences are ignored, the program generates feasible sequential cutting-and-welding patterns with moderate running times for moderate  $K$ s and possibly large running times for large  $K$ s (see column G4 in Table 2). If decision maker preferences are taken into consideration, the program generates minimal sequential cutting-and-welding patterns with running times varying from (milli)seconds to hours (see pattern J in Table 2).

4.4.2. *Reformulating the Pattern Conversion Problem for Sequential Cutting-and-Welding Patterns and Decision Maker Preferences.* For single-size stocks (6 meters, as in the case which motivates this article, or different), sequential cutting-and-welding patterns may also be generated by formulating an assignment-like problem, a permutation of pipes, where  $x_{\ell j} \in \{0, 1\}$  represents the place of pipe  $\ell$  in the permutation and where the objective function is arbitrary, because generating one sequential pattern is sufficient:

$$\begin{aligned} \sum_{\ell=1}^L x_{\ell j} &= 1 \quad j = 1 \dots L \\ \sum_{j=1}^L x_{\ell j} &= 1 \quad \ell = 1 \dots L \\ \sum_{j=1}^{m-1} \sum_{\ell=1}^L s_{\ell} x_{\ell j} &\geq k_m 6000 \quad m = 1 \dots L \\ \sum_{j=1}^m \sum_{\ell=1}^L s_{\ell} x_{\ell j} &\leq (k_m + 2) 6000 \quad m = 1 \dots L \\ x_{\ell j} &\in \{0; 1\} \quad \ell = 1 \dots L, j = 1 \dots L \\ k_m &\in \mathbb{N} \quad m = 1 \dots L. \end{aligned} \tag{6}$$

The  $\sum_{j=1}^{m-1} \sum_{\ell=1}^L s_{\ell} x_{\ell j} \geq k_m 6000$  constraints ensure that the aggregate size of the first  $m - 1$  pipes in the permutation is

at least  $k_m 6000$ ; in other words, there are at least  $k_m$  stocks for the first  $m - 1$  pipes. The  $\sum_{j=1}^m \sum_{\ell=1}^L s_{\ell} x_{\ell j} \leq (k_m + 2) 6000$  constraints ensure that the aggregate size of the first  $m$  pipes in the permutation is  $(k_m + 2) 6000$  at the most; in other words, as well as the leftover segment from stock  $k_m$ , an additional stock may be used for producing pipe  $m$ . These two sets of constraints ensure that pipe  $m + 1$  in the permutation has one weld at the most and that the program generates a minimal sequential cutting-and-welding pattern, albeit one which disregards decision maker preferences (see column S1 in Table 2). The program running times are moderate for all patterns with the exception of J, which does not have a minimal integer solution.

Decision maker preferences may be taken into consideration, but the pattern conversion problem formulated as a mixed integer linear program is more cumbersome; if  $W_j \in \{0; 1\}$  is the binary variable indicating whether the  $j$ th pipe in the permutation is welded or not and  $w_{\ell} \in \{0; 1\}$  is the binary variable which shows whether the  $\ell$ th pipe in the original indexing is welded or not, then

$$\begin{aligned} \sum_{\ell=1}^L o_{\ell} w_{\ell} &\longrightarrow \min \\ \text{s.t.: } \sum_{\ell=1}^L x_{\ell j} &= 1 \quad j = 1 \dots L \\ \sum_{j=1}^L x_{\ell j} &= 1 \quad \ell = 1 \dots L \\ \sum_{j=1}^{m-1} \sum_{\ell=1}^L s_{\ell} x_{\ell j} &\geq k_m 6000 \quad m = 1 \dots L \\ \sum_{j=1}^m \sum_{\ell=1}^L s_{\ell} x_{\ell j} &\leq (k_m + 1) 6000 + W_m 6000 \\ & \quad m = 1 \dots L \\ \sum_{\ell=1}^L W_{\ell} &= K - 1 \\ x_{\ell j} + W_j &\leq w_{\ell} + 1 \quad \ell = 1 \dots L, j = 1 \dots L \\ x_{\ell j} &\in \{0; 1\} \quad \ell = 1 \dots L, j = 1 \dots L \\ k_m &\in \mathbb{N} \quad m = 1 \dots L \\ W_m &\in \{0; 1\} \quad m = 1 \dots L \\ w_m &\in \{0; 1\} \quad m = 1 \dots L \end{aligned} \tag{7}$$

The  $\sum_{j=1}^m \sum_{\ell=1}^L s_{\ell} x_{\ell j} \leq (k_m + 1) 6000 + W_m 6000$  constraint ensures that pipe  $k_m$  in the permutation is welded if  $W_m = 1$ . The ‘original’ index of pipe  $k_m$  is unknown, but the  $x_{\ell j} + W_j \leq w_{\ell} + 1$  constraint ensures that pipe  $\ell$  is welded if pipe  $\ell$  is in the  $j$ th place in the permutation and the  $j$ th pipe in the permutation is welded.

Unfortunately, the algorithm running times turn out to be large, longer than seconds just to generate a feasible solution (see column G5 in Table 2).

In summary, sequential cutting-and-welding patterns may be generated in simple circumstances and possibly complex ones too, while decision maker preferences may be taken into account in simple circumstances, but not in complex ones.

## 5. Results and Discussion

The algorithm suggested in this article (see Section 4.1) was applied to the case of a Hungarian manufacturer of fixed firefighting systems (see Section 4.4). The company uses single-size, 6-meter stocks (see Table 1 in Section 1 for illustrative pipe sizes and quantities), the stock price is  $\pi_1$ , and the welding price  $\pi_w$  is generally low. However, increases in demand lead to increases in both welding time and opportunity cost, which in turn lead to increases in welding price.

The software used—BarCut (version 2.1)—was suitable for both single- and multiple-size stocks. The number of stocks per cutting-and-welding pattern was limited to  $K = 10$  and, to reflect the opportunity cost,  $\pi_w$  was set as a percentage (90%, 49%, and 30%) of  $\pi_1$ , assumed invariable; Tables 3, 4, and 5 show the minimal cutting-and-welding patterns for the three welding prices.

When  $\pi_w$  is high ( $\pi_w = 0.9\pi_1$ ), welding is used only when inevitable, for pipes longer than 6 meters. To generate the optimal equivalent cutting plan, the program running time is 8.45 seconds and involves 30 stocks and 4 welds (see Table 3).

When  $\pi_w$  is moderate ( $\pi_w = 0.49\pi_1$ ), welding is used less restrictively. To generate the optimal equivalent cutting plan, the program running time is 8.47 seconds and involves 29 stocks and 6 welds (see Table 4).

When  $\pi_w$  is low ( $\pi_w = 0.3\pi_1$ ), welding is used more liberally. To generate the optimal equivalent cutting plan, the program running time is 8.76 seconds and involves 27 stocks and 12 welds (see Table 5). Since the aggregate length of the leftover segments is just 5.034 meters, lower welding prices and further investigation—with  $\pi_w < 0.3\pi_1$ —would be useless.

Optimal equivalent cutting plans need to be converted to minimal cutting-and-welding patterns. Since all the pipes in these optimal equivalent cutting plans were either shorter than 6 meters or involved only two stocks, the conversions are easy; more complex cases were analysed in Sections 4.2 and 4.4.

## 6. Conclusions

Due to its practical relevance across the entire industrial spectrum, the cutting-stock problem continues to be studied intensively in operations research and to generate sophisticated algorithms solving it. Some of these were translated into open-source or commercial programs, catering to the diverse user needs and circumstances, at various prices and under various terms and conditions.

TABLE 3: Minimal cutting-and-welding patterns for  $\pi_w = 0.9\pi_1$ .

Pattern	Stock quantities	Weld numbers	Pipe sizes	Repeats
1	1	0	2000+4000	1
2	1	0	4500	2
3	1	0	4680	11
4	1	0	5000	1
5	1	0	5096	1
6	1	0	5250	1
7	1	0	5260	1
8	1	0	5500	1
9	1	0	5660	1
10	1	0	6000	2
11	2	1	7200+4500	2
12	2	1	7000+4660	2

TABLE 4: Minimal cutting-and-welding patterns for  $\pi_w = 0.49\pi_1$ .

Pattern	Stock quantities	Weld numbers	Pipe sizes	Repeats
1	1	0	5260	1
2	1	0	2000+4000	1
3	3	2	4500+4500+4500+4500	1
4	2	1	4680+7200	2
5	2	1	5000+7000	1
6	1	0	4680	8
7	1	0	5096	1
8	1	0	5500	1
9	1	0	5660	1
10	1	0	6000	2
11	2	1	4680+7000	1
12	1	0	5250	1
13	1	0	4660	2

Since the first cutting-stock problems studied in operations research were one-dimensional, increasing dimensionality and diversifying methodology became natural extensions. This article is one of only a handful to focus on the effect of welding on the one-dimensional cutting-stock problem instead; it is also the first of its kind to limit the number of welds to one, to conform to best practice safety requirements for fixed firefighting systems in the construction industry.

This article defined the cutting-stock problem with welding (see Section 3.2) and argued that it can be converted to an equivalent cutting-stock problem with multiple-size stocks (see Section 4.1), a problem that can be solved with existing algorithms and software (see Section 3.1). For probably the first time in the literature, it also defined the pattern conversion problem, whereby cutting patterns are converted to cutting-and-welding patterns as a mixed integer linear program.

TABLE 5: Minimal cutting-and-welding patterns for  $\pi_w = 0.3\pi_1$ .

Pattern	Stock quantities	Weld numbers	Pipe sizes	Repeats
1	1	0	6000	2
2	2	1	4680+7200	2
3	1	0	2000+4000	1
4	1	0	5660	1
5	1	0	5500	1
6	1	0	5260	1
7	1	0	5250	1
8	2	1	5000+7000	1
9	4	3	4660+4660+4680+4680+4680	1
10	2	1	4680+7000	1
11	3	2	4500+4500+4500+4500	1
12	4	3	4680+4680+4680+4680+4680	1
13	1	0	5096	1

Optimal solutions to the bounded cutting-stock problems with welding were investigated with the algorithm suggested in this article and constraints were examined (see Section 4.1). The implications for unbound cutting-stock problems with welding could not be ascertained (see Section 4.3); however, the algorithm was found to generate a heuristic solution at the least (see Section 4.4).

Converting equivalent cutting patterns to cutting-and-welding patterns is fundamental to the algorithm suggested in this article. For sequential cutting-and-welding patterns and decision maker preferences, this pattern conversion problem was (re)formulated as a mixed integer linear program (see Section 4.4.2).

The cutting-stock problem with welding studied in this article was one-dimensional; increasing dimensionality is a natural extension. However, two- and three-dimensional cutting-stock problems with welding present further complexities which require further research and algorithms different than the one suggested in this article; more likely than not, large instances cannot be solved to optimality.

## Data Availability

The data used to support the findings of this study are included within the article.

## Disclosure

Preliminary version has been presented at 20th European Conference on Mathematics for Industry and at EURO/ALIO 2018, International Conference on Applied Combinatorial Optimization.

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

Kolos Ágoston acknowledges the support of the Hungarian Academy of Sciences under its Cooperation of Excellences Grant (KEP-6/2017). The author is very grateful to colleagues at the Department of Operational Research and Actuarial Sciences, Corvinus University of Budapest, for relevant comments and suggestions on the earliest draft, to Péter Biró and Ágnes Cseh at the Mechanism Design Research Unit, Institute of Economics, Centre for Economic and Regional Studies, Hungarian Academy of Sciences, for precious comments and suggestions on another intermediary draft, and to Anamaria M. Cristescu-Martin and Márton Benedek, for editorial assistance with many drafts since.

## References

- [1] L. V. Kantorovich, *Mathematical Methods of Organizing and Planning Production*, vol. 6, Publication House, Leningrad State University, Leningrad, Russia, 1939.
- [2] L. V. Kantorovich, "Mathematical methods of organizing and planning production," *Management Science*, vol. 6, no. 4, pp. 366–422, 1960.
- [3] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting-stock problem," *Operations Research*, vol. 9, no. 6, pp. 849–859, 1961.
- [4] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting-stock problem—part ii," *Operations Research*, vol. 11, no. 6, pp. 863–888, 1963.
- [5] P. E. Sweeney and E. R. Paternoster, "Cutting and packing problems: a categorized, application-orientated research bibliography," *Journal of the Operational Research Society*, vol. 43, no. 7, pp. 691–706, 1992.
- [6] G. M. Roodman, "Near-optimal solutions to one-dimensional cutting stock problems," *Computers & Operations Research*, vol. 13, no. 6, pp. 713–719, 1986.
- [7] G. Wäscher and T. Gau, "Heuristics for the integer one-dimensional cutting stock problem: a computational study," *OR Spectrum*, vol. 18, no. 3, pp. 131–144, 1996.

- [8] M. Gradisar and P. Trkman, "A combined approach to the solution to the general one-dimensional cutting stock problem," *Computers & Operations Research*, vol. 32, no. 7, pp. 1793–1807, 2005.
- [9] H. Dyckhoff, "A new linear programming approach to the cutting stock problem," *Operations Research. The Journal of the Operations Research Society of America*, vol. 29, no. 6, pp. 1092–1104, 1981.
- [10] G. Scheithauer and J. Terno, "A branchbound algorithm for solving one-dimensional cutting stock problems exactly," *Appliationes Mathematicae*, vol. 23, no. 2, pp. 151–167, 1995.
- [11] P. H. Vance, C. Barnhart, E. L. Johnson, and G. L. Nemhauser, "Solving binary cutting stock problems by column generation and branch-and-bound," *Computational Optimization and Applications*, vol. 3, no. 2, pp. 111–130, 1994.
- [12] F. Vanderbeck, "Computational study of a column generation algorithm for bin packing and cutting stock problems," *Mathematical Programming*, vol. 86, no. 3, pp. 565–594, 1999.
- [13] G. Belov and G. Scheithauer, "A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting," *European Journal of Operational Research*, vol. 171, no. 1, pp. 85–106, 2006.
- [14] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130, 2007.
- [15] J. Levine and F. Ducatelle, "Ant colony optimization and local search for bin packing and cutting stock problems," *Journal of the Operational Research Society*, vol. 55, no. 7, pp. 705–716, 2004.
- [16] S. A. Araujo, A. A. Constantino, and K. C. Poldi, "An evolutionary algorithm for the one-dimensional cutting stock problem," *International Transactions in Operational Research*, vol. 18, no. 1, pp. 115–127, 2011.
- [17] Y. Cui and Y. Yang, "A heuristic for the one-dimensional cutting stock problem with usable leftover," *European Journal of Operational Research*, vol. 204, no. 2, pp. 245–250, 2010.
- [18] C. H. Cheng, B. R. Feiring, and T. C. E. Cheng, "The cutting stock problem - a survey," *International Journal of Production Economics*, vol. 36, no. 3, pp. 291–305, 1994.
- [19] C. Goulimis, "Optimal solutions for the cutting stock problem," *European Journal of Operational Research*, vol. 44, no. 2, pp. 197–208, 1990.
- [20] E. J. Zak, "The skiving stock problem as a counterpart of the cutting stock problem," *International Transactions in Operational Research*, vol. 10, no. 6, pp. 637–650, 2003.
- [21] J. Martinovic and G. Scheithauer, "Integer linear programming models for the skiving stock problem," *European Journal of Operational Research*, vol. 251, no. 2, pp. 356–368, 2016.
- [22] D. Tanir, O. Ugurlu, A. Guler, and U. Nuriyev, "One-dimensional cutting stock problem with divisible items," 2016.