

One-Dimensional Cutting Stock Optimisation by Suborders

Mirko GRADIŠAR, Mihael CESAR, Luka TOMAT

Abstract: This paper introduces a method for solving a one-dimensional cutting stock problem by suborders. The method is used for large orders that for technological and logistical reasons cannot be filled in a single order, but only in several successive suborders. The method has two stages. In the first stage, the suborders are generated and in the second the trim-loss is minimised. All leftovers longer than D are returned to stock and reused. Shorter leftovers are treated as trim-loss and discarded. A detailed description of the method is provided by using a practical case. The method is tested by solving 108 randomly generated problem instances.

Keywords: algorithm; cutting stock problem; discrete optimisation; large order

1 INTRODUCTION

Material cutting can be studied from different viewpoints [1]. One of them is trim-loss minimisation in one-dimensional stock cutting [2]. The cutting stock problem (CSP) occurs in many industries. Examples range from the metal [3], paper [4], textile and wood industries [5] through to nanotechnology [6]. They face the same problem of how to cut ordered items out of what are generally different stock lengths to minimise the trim-loss [7] and maximize the utilization of stocks [8]. The trim-loss is the leftovers of the cutting process that are too short to be used in subsequent orders and are treated as waste.

Reducing the trim-loss is typically the main objective of any optimisation, but it is not the only one. Other objectives and restrictions must be taken into consideration. They vary from case to case due to a combination of technical, production and business characteristics that are additional constraints on the model and which decrease the number of possible solutions. However, the time needed to find the optimal solution, which is defined as a global minimum of criteria function, can be too long in large-scale problems and may create a bottleneck in the business process. The reason for the time-consuming calculation of the optimal solution is the fact that most CSPs are NP-complete [7]. Therefore, only a few papers describing an exact solution of a small-scale CSP can be found in the literature [9]. Most authors propose a heuristic approach which leads to acceptable, yet not always optimal results.

Companies around the world encounter various kinds of CSPs in which standard methods are inappropriate for solving them, and therefore new or modified methods need to be developed. Standard methods generally minimise the level of trim-loss. Many companies try to achieve other objectives as well, such as a reduction of lead times, the minimisation of cutter setting changes, a reduction of machine degradation time [10], a lowering of stocks, adjusting the cutting procedure to other business processes, and similar [11].

One specific problem that cannot be solved with standard methods is the optimisation of large-scale orders. Large-scale orders with a high number of stock and order lengths cannot be processed in a single session due to technological or logistical limitations and therefore need to be divided into suborders. A standard method could be

used if suborders were to be formed after optimisation of the entire order. However, in some cases this is impossible without violating additional constraints regarding suborders, for instance that each stock length can only be used in one suborder. In such cases, first the suborders are created and then the CSP for each of them is solved. Another possible reason that standard methods cannot be used is that the problem is so large that it exceeds the limitations of a certain computer application such as the maximum number and pieces of order and stock lengths.

In the literature we were unable to find a paper that describes a solution to a similar problem even though it exists in practice. Therefore, we propose a method for solving a CSP for a large order by dividing it into smaller parts, or suborders, in order to achieve the minimum total trim-loss for the entire order. The proposed method is developed for a specific practical situation but it can also be adapted for use in other organisations with such a problem since there are only a few specific constraints.

This paper has five sections. Section 2 defines the problem. Section 3 develops the solution to it. Section 4 tests the method by using real data and randomly generated problem instances. Finally, Section 5 presents the conclusion and outlines further research possibilities.

2 PROBLEM DEFINITION

A sufficient amount of one-dimensional material is found in stock (stock lengths). Stocks can be of different sizes in a varying number of pieces and are expressed in integers. They differ because a manufacturer produces a variety of different standard lengths. It should be noted that important differences exist between the declared standardised lengths and the actual measured ones. In addition, there are non-standardised lengths, which are usually leftovers from previous orders.

Material held in stock is used to fill orders of various shorter lengths in a specified number of items. Due to its excessive size, each order is divided into r suborders. The described problem is reversed with regard to combining various orders with the aim of filling them before a certain deadline [12] and is similar to the CSP in consecutive periods in which multiple orders need to be processed [13], except that in our case the stock assortment is fixed while processing the order. The order characteristics are known in advance and are not stochastic, as in [14].

The material left after the cutting procedure is returned to stock if its size is larger than a certain threshold [9]. A one-dimensional CSP in which leftovers can be reused is denoted as a 1DCSPUL (One Dimensional Cutting Stock Problem with Usable Leftovers). Leftovers larger than a threshold D are defined as usable leftovers (UL) and shorter ones are trim-loss. The order is filled in such a way that the total trim-loss of all suborders is minimised. The solution must take the following requirements into account:

- 1) all items of a certain order length are placed in only one suborder;
- 2) each stock length is used in only one suborder;
- 3) each suborder consists of maximum P order lengths;
- 4) each stock length can be cut to maximum of N order lengths; and
- 5) the size of the suborder is limited to M and is defined as the sum of all pieces ordered in that suborder.

All five of the above requirements are the result of specific business processes that depend on material cutting in a large international company that manufactures towers for high-voltage transmission lines. A deep analysis of those processes would exceed the scope of this paper, although very similar constraints can probably also be found in other companies facing the problem of large-scale orders.

We designed the criteria function and its constraints according to the specifics of the enterprise. The cutting stock problem consists of two stages. In the first stage, leftovers are minimised and in the second one the trim-loss is. The problem is defined as follows:

l_i - order lengths; $i = 1, \dots, n$

b_i - required number of pieces of order length l_i

L_j - stock lengths; $j = 1, \dots, m$

x_{ij} - number of pieces of order length l_i that are cut from stock length j

δ_{1j} - leftover after cutting stock length L_j in the first stage

δ_{2j} - leftover after cutting stock length L_j in the second stage

Criteria function for the first stage:

$$\min \sum_{k=1}^r \sum_{j=1}^m t_{kj} \text{ (minimisation of leftovers for the entire order)}$$

Subject to:

$$1) \sum_{i=1}^n l_i b_i h_{ki} \leq M, \forall k \text{ (sum of order lengths used in suborder } k)$$

$$2) \sum_{k=1}^r g_{1kj} \leq 1, \forall j \text{ (each stock length can be used in only one suborder)}$$

$$3) \sum_{k=1}^r h_{ki} \leq 1, \forall i \text{ (each order length can be used in only one suborder)}$$

$$4) \sum_{i=1}^n h_{ki} \leq P, \forall k \text{ (maximum number of order lengths in one suborder)}$$

$$5) \sum_{i=1}^n y_{ij} \leq N, \forall j \text{ (one stock length can be cut into } N \text{ different order lengths)}$$

$$6) \sum_{i=1}^n l_i x_{ij} + \delta_{1j} = L_j, \forall j \text{ (knapsack constraints)}$$

$$7) \sum_{j=1}^m x_{ij} = b_i, \forall i \text{ (order lengths are cut into the exact number of pieces required)}$$

$$8) \sum_{j=1}^m z_j u_{kj} \leq 1, \forall k \text{ (only one leftover in suborder } k \text{ can be longer than } \max(l_i h_{ki}) \text{ in order to prevent the excessive generation of usable leftovers)}$$

$$x_{ij} \geq 0, \text{ integer}, \forall i, j$$

$$t_{kj} \geq 0, \forall j$$

$$\delta_{1j} \geq 0, \forall j$$

In the first stage, the following variables are used: $z_j = 0$ if $x_{ij} = 0 \forall i$; 1 otherwise to indicate whether stock length L_j is used in the cutting plan, $g_{1kj} = 1$ if L_j is used in suborder k ; 0 otherwise to indicate whether stock length L_j is used in suborder k in the first stage, $h_{ki} = 1$ if l_i is used in suborder k ; 0 otherwise to indicate whether order length l_i is used in suborder k , $u_{kj} = 1$ if $g_{1kj} = 1 \wedge \delta_j > \max(l_i h_{ki})$; 0 otherwise to indicate if the leftover of L_j , used in suborder k , is larger than $\max(l_i h_{ki})$, $t_{kj} = \delta_{1j}$ if $g_{1kj} = 1$; 0 otherwise t_{kj} is trim-loss related to L_j in suborder k , $y_{ij} = 0$ if $x_{ij} = 0$; 1 otherwise to indicate if order length l_i is cut from stock length L_j .

The results of the first stage are minimal leftovers and corresponding number of used stock lengths B . The threshold for trim-loss D is not included in the first-stage model. As only one leftover per suborder is allowed to exceed $\max(l_i h_{ki})$, in the scenario in which D would be set below $\max(l_i h_{ki})$, the following situation can potentially occur. In cases where the stock is large enough, the algorithm would generate as many leftovers as possible with their length less than $\max(l_i h_{ki})$ and greater than D . The trim-loss would be low but would come at the cost of the excessive growth of UL. Too many UL returned to stock increase the costs of logistics and warehousing. The same scenario would emerge if more than one leftover per suborder larger than $\max(l_i h_{ki})$ is permitted. Because the maximum one leftover is greater than $\max(l_i h_{ki})$ (8), it would make no sense to set a larger D . For this reason, the only possible value for D in the k^{th} group would be $\max(l_i h_{ki})$. Since D primarily depends on the cost of material, logistics and warehousing and does not depend on the group, this would not be acceptable. Therefore, the decision-maker set the threshold for UL at D after the leftover minimisation is carried out and sometimes in dependence on its results.

When the leftover minimisation is completed and D is set, the second stage – trim-loss minimisation – can start. The trim-loss minimisation problem is defined as follows:

Criteria function for the second stage:

$\min \sum_{k=1}^r \sum_{j=1}^m w_{kj}$ (minimisation of the trim-loss for the entire order)

Subject to:

9) $\sum_{k=1}^r \sum_{j=1}^m g_{2kj} = B$ (the number of stock lengths used is the same as in the first stage).

The constraints are also the same as in first stage, except (6) where δ_{1j} is replaced with δ_{2j} and (8) which is replaced by (9).

In the second stage, three additional variables are used:

$$B = \sum_{k=1}^r \sum_{j=1}^m g_{1kj} .$$

Where B is the number of stock lengths

used in the first stage, $g_{2kj} = 1$ if L_j is used in suborder k ; 0 otherwise to indicate whether stock length L_j is used in suborder k in the second stage, $w_{kj} = \delta_{2j}$ if $g_{2kj} = 1 \wedge \delta_{2j} \leq D$; 0 otherwise w_{kj} is the trim-loss related to L_j in suborder k in the second stage.

It is assumed that the manipulation and logistic costs do not depend on the length of the stock objects. In the second stage there can thus be more than one leftover larger than $\max(l_i h_{ki})$ since the excessive generation of UL is prevented by constraint (9). Shorter stock lengths with a larger trim-loss from the first stage could in the second one be replaced with larger ones and so the trim-loss would be transformed to UL by keeping the same number of used stock lengths B .

Although the described problem reflects a specific practical situation, it is only different from the general CSPUL definition [7] in constraints concerning the suborder size limit (1) and (4) and in the two additional constraints (3) and (2). The most important is constraint (3) because if constraint (2) is omitted it would not fundamentally influence the proposed method that is introduced in the next section. Therefore, it is expected that the method could also be easily adapted for use in other companies dealing with large-scale orders.

3 SOLUTION DEVELOPMENT

Suborders may be generated in two ways. The order can be optimised first as a whole and the results can be later divided into smaller suborders. But, in this case, constraint (3) cannot be met. If any stock length can be cut to different order lengths, then it is generally impossible to group stock lengths that produce only certain order lengths. The second way therefore needs to be applied in our case where suborders are generated before the cutting plan of each suborder is calculated.

The proposed method is developed for large-scale orders where there is not only a large number of ordered pieces but also where a large number of pieces are needed from stock to fill the order. This means that the order lengths are large compared to the stock lengths and, on average, only one to three ordered pieces are expected to be cut from a single stock length. The method is thus adapted for the kind of CSP that can most frequently be expected in practice when dealing with large-scale orders.

The presented problem would be very difficult to implement in some mathematical modelling programming language. This would also make no sense since the exact solution in the case of large-scale orders cannot be found in an acceptable time period. Accordingly, a heuristic method needs to be developed.

The proposed method has two stages. The first stage divides the large-scale order into suborders and minimises the leftovers. It has two phases. In the first phase, pairs are formed that represent the smallest possible suborders. In the second phase, the final suborders are formed and the leftovers of suborders are minimised. In the second stage some of the trim-loss is transformed into UL.

The method is based on the assumption that starting the cutting procedure with the longest ordered pieces yields better results by minimising the effect of "ending conditions" [6]. For this purpose, in the first phase the set of order lengths S_n is formed and sorted in descending order. After sorting, the first one, the longest order length, is selected and the most suitable pair for it is determined. If set S_n has an odd number of lengths, the number is artificially made even by adding one more order length in one piece with a length *maxint*. *maxint* is the largest possible integer number that can be represented in the computer. With single precision, this number equals 32.767. Since such a large order piece cannot be processed, it does not have any influence on the result. However, in this way every order length is guaranteed to have a pair.

A pair is formed by solving the knapsack problem for all possible combinations for the largest order length from list S_n with the other shorter ones, while considering all available stock. The pair with the lowest average trim-loss is chosen. The average trim-loss is calculated as a ratio of the sum of leftovers of all stock objects used decreased by the total length of leftovers larger than $\max(l_i h_{ki})$ and the sum of stock pieces used decreased by leftovers. After choosing the optimal pair, all of the used order and stock lengths are excluded from further calculation. The procedure is repeated until list S_n is empty. The result of this procedure is $n/2$ pairs sorted by descending average trim-loss.

Instead of pairs, one could also form groups of three or more order lengths without the need to essentially change the procedure. The decision to use pairs was made because in large orders it can be expected that in most cases only one, two or exceptionally three order pieces can on average be cut from one stock length. More than two order lengths would rarely be used especially when several pieces of the same order length are combined. The other reason for pairs is the lowest number of possible combinations.

The obtained sorted pairs form the set S_r . In the second phase, S_r is divided into suborders moving from the top of the set towards the end. The number of pairs in each suborder is limited by P and M .

The resulting suborders are now independent but they all are using the same stock. Therefore, it is essential that at the end of the second phase the suborders are optimised in the same order as they were created. Thus, for the first suborder with the largest average trim-loss all stock lengths are available, meaning more combinations are possible and a potential opportunity to lower the trim-loss [7]. To optimise the second and other suborders down the line the material in stock that is left over from the previous one can

be used. The number of possible combinations is hence reduced but, on the other side, the remaining suborders consist of pairs that are easier to combine in an acceptable solution.

To minimise the leftovers of a particular suborder, any existing method for solving the 1DCSPUL can be selected. In our case, the latest version of the COLA 4] computer application was used because it is able to process large orders with up to 900 order and 900 stock pieces. The CPLEX solver was added for the exact solution of small suborders and the LCUT application for cases of large order lengths.

The result of the first stage is the cutting plan. In each suborder at most one leftover is larger than $\max(l_i h_{ki})$ (8). The decision-maker decides about D and leftovers of all suborders larger than D can then be treated as UL and returned to stock and so the procedure can be completed. But in cases where at least a few larger objects are left unused in stock, the trim-loss could be further substantially reduced by applying the second stage of the proposed method.

In the second stage, the UL can be increased on account of the trim-loss. The trim-loss is lowered in a manner that does not consume a greater number of stock lengths and so further reduce the criteria function. With these stock lengths, shorter ones with a high trim-loss are replaced. The result is leftover higher than D . The high trim-loss is thus transformed into UL. Fig. 1 describes both stages and phases of the proposed algorithm in pseudo code.

FIRST STAGE

Determination of pairs

Sort order lengths descending to

$S_n(l_1 > l_2 > \dots > l_i > \dots > l_n)$.

$j \leftarrow 1, \dots, n-1; k \leftarrow 2, \dots, n$

$R_{j,k} \leftarrow \maxint \forall j, k$ (starting values of the average trim-loss for all pairs)

$i \leftarrow 1$ (starting value of counter i)

$h \leftarrow 1$ (starting value of pairs' index with minimum trim-loss)

while $i < n-1$

$p \leftarrow i+1$ (starting value of counter p)

$R_{\min} \leftarrow \maxint$ (starting value of minimum trim-loss)

$c \leftarrow 0$ (starting value of test variable c)

while $p < n$

if $l_i > 0 \wedge l_p > 0$ **then**

do KNAPSACK¹ for l_i
and l_p .

compute average trim-loss R .

if $R < R_{\min}$ **then**

$R_{\min} \leftarrow R$

$j \leftarrow i$

$k \leftarrow p$

$c \leftarrow 1$ (c tests if at least once $R < R_{\min}$ was true)

end if

end if

$p \leftarrow p+1$

end while

if $c = 1$ **then**

$R_h \leftarrow R_{\min}$

$s1_h \leftarrow j$

$s2_h \leftarrow k$

$l_j \leftarrow 0; l_k \leftarrow 0$ (exclude consumed stocks for pair l_j and l_k)

$h \leftarrow h+1$

end if

$i \leftarrow i+1$

end while

Creation of suborders and solution of the 1DCSPUL

Sort average trim-losses R_h and belonging pair numbers descending by the value R_h to the sets $S_1(R_1 > R_2 > \dots > R_{n/2})$, $S_{p1}(s1_1, s1_2, \dots, s1_{n/2})$, and $S_{p2}(s2_1, s2_2, \dots, s2_{n/2})$.

$h \leftarrow 1$ (starting value of the pair counter)

$a \leftarrow 1$ (present the number of different order lengths in the suborder)

$b \leftarrow 0$ (present the total length of order pieces in the suborder)

$g1 \leftarrow 1$ (present the lower bound of the suborder)

while $h \leq n/2$

$i \leftarrow g1$

while $i \leq h$

$a \leftarrow a+2$

$b \leftarrow l_{s1i} b_{s1i} + l_{s2i} b_{s2i}$

$i \leftarrow i+1$

end while

if $a > P \vee b > M$ **then**

$g2 \leftarrow i-2$ (present the upper bound of the suborder)

if $g1 < g2$ **then**

$g1 \leftarrow g2$ (if one pair exceeds P or M , it is cut anyway)

end if

1DCSPUL optimisation for the suborder of pairs $s1_{g1}$ to $s1_{g2}$ and $s2_{g1}$ to $s2_{g2}$ with the COLA algorithm.

$g1 \leftarrow g2+1$

$h \leftarrow h-1$

end if

$h \leftarrow h+1$

end while

SECOND STAGE

Transformation of the trim-loss into UL

sort δ_{ij} in a non-increasing manner to W ($W_1 \geq W_2 \geq \dots \geq W_m$) and **make** L_q ($L_{q1}, L_{q2}, \dots, L_{qm}$) of corresponding stock lengths L_j .

sort $(1 - g1_{kj}) \cdot L_j$ in an increasing manner to L_n ($L_{n1} \leq L_{n2} \leq \dots \leq L_{nm}$).

$h \leftarrow 1$ (trim-loss counter)

$k \leftarrow 1$ (unused bars counter)

¹The KNAPSACK procedure is the same as in [6].

```

while  $h \leq m$ 
  while  $k \leq m$ 
    if  $(L_{nk} - L_{qh}) > (D - W_h)$  then
       $W_h \leftarrow W_h + L_{nk} - L_{qh}(W_h)$ 
      is greater than  $D$  and
      becomes a usable leftover)
       $a \leftarrow L_{qh}; L_{qh} \leftarrow L_{nk}; L_{nk} \leftarrow a$  (replace
      stock lengths from  $L_n$  and  $L_q$ )
    end if
     $k \leftarrow k + 1$ 
  end while
   $h \leftarrow h + 1$ 
   $k \leftarrow 1$ 
end while
    
```

Figure 1 Algorithm for the cutting stock process by suborders in pseudo code

Without constraint (8) in which only one leftover larger than $\max(l_i, h_{ki})$ is allowed, optimised pairs could simply be gathered into suborders.

The proposed heuristic method is implemented in the form of the S-CUT (Suborder Cutting) computer program that includes the COLA application. The limitations of S-CUT are the same as those of COLA.

4 RESULTS

The optimisation of a practical example of cutting steel bars is shown in order to illustrate the use of S-CUT. The example has the following constraints: $P = 8, N = 2, M = 150000$ and $D = 2500$. Tab. 1 presents the input data.

Order lengths are sorted from the longest to the shortest, which is in line with the proposed method.

Table 1 Stock and order lengths

INVENTORY		ORDER	
Stock length (mm)	Number of pieces	Order length (mm)	Number of pieces
12965	7	9940	4
11965	10	9450	2
10965	37	8480	2
6945	2	7530	2
6465	4	6910	4
		6145	2
		6000	4
		5710	2
		5600	12
		5280	8
		4825	12
		420	18

The example shows that the total length of all ordered pieces amounts to 52% of the material in stock. The order lengths are large since the ratio between the average stock and order length is 1.55. This means that, on average, fewer than two order pieces are cut from a single stock length.

The result of the first phase is a list of pairs in descending order by their average level of trim-loss. Because there are 12 different order lengths, there are six pairs. Since $P = 8$, the pairs are divided into two suborders which are presented in Tab. 2.

The first suborder includes four pairs with eight different order lengths while the second suborder includes the last two pairs with four order lengths. In the second phase, S-CUT first optimises the first suborder with pairs that have higher levels of average trim-loss. The whole

stock is available for the first suborder. The rest of the stock lengths are used to optimise the second suborder. The results of the first stage are shown in Tabs. 3 and 4.

Table 2 Division of pairs into suborders

Pair. no	First order length	Second order length	Average trim-loss
Suborder 1			
1	7530	4825	1113
2	8480	5280	1098
3	6000	5600	565
4	9450	6910	528
Suborder 2			
5	9940	420	135
6	6145	5710	101

Table 3 Cutting plan for suborder 1 (after the first stage)

DATA			
INVENTORY		ORDER	
Stock length	Number of pieces	Order length	Number of pieces
12965	7	9450	2
11965	10	8480	2
10965	37	7530	2
6945	2	6910	4
6465	4	6000	4
		5600	12
		5280	8
		4825	12
CUTTING PLAN			
Stock length	Number of pieces	Cutting patterns	Leftover
6945	2	6910	35
10965	8	5280; 5600	85
10965	4	4825; 6000	140
11965	2	4825; 6910	230
10965	4	4825; 5600	540
12965	2	7530; 4825	610
10965	2	9450	1515
10965	2	8480	2485

The results for both suborders show that no trim-loss is longer than $\max(l_i, h_{ki})$, which is 9450 mm for suborder 1 and 9940 mm for suborder 2. All leftovers count as a trim-loss because they are all shorter than 2500 mm.

By carrying out the second-stage procedure, the two consumed stock lengths initially 10965 mm long and with a trim-loss of 2485 mm have been replaced by two unused ones 11965 mm in length. The leftover rises to 3485 mm and becomes UL. Similarly, the trim-loss of 1515 mm is changed to 2515 mm. Three unused stock lengths 12965 mm long replaced three of four consumed ones that were initially 10965 mm long and with a trim-loss 540 mm which is now changed to 2540 mm. At the end, two unused stock lengths 10965 mm in length replaced the two consumed ones initially 6945 mm long and with a trim-loss of 35 mm. The leftover is thereby extended to 4055 mm. All other trim-loss is too short to be extended over the threshold of 2500 mm. The result of the second stage is a new cutting plan of suborder 1 that is presented in Table 5. Four leftovers are longer than 2500 mm and are treated as UL. The total trim-loss is lowered from 14010 mm to 4220 mm, representing a reduction of 70%.

The example illustrated in this section was provided by a company; nevertheless, it is considered to be amongst the smallest. It was chosen because an average order is at least 10-times larger and would take too much space here to present it without providing any additional information

about the method. The processing time was less than a second on a personal computer.

Table 4 Cutting plan for suborder 2 (after the first stage)

DATA			
INVENTORY		ORDER	
Stock length	Number of pieces	Order length	Number of pieces
12965	5	9940	4
11965	8	6145	2
10965	17	5710	2
6465	4	420	18
CUTTING PLAN			
Stock length	Number of pieces	Cutting patterns	Leftover
11965	2	6145; 5710	110
12965	2	9940; 7 × 420	85
10965	2	9940; 2 × 420	185

Table 5 Cutting plan for suborder 1 (after the second stage)

CUTTING PLAN			
Stock length	Number of pieces	Cutting patterns	Leftover
10965	2	6910	4055
10965	8	5280; 5600	85
10965	4	4825; 6000	140
11965	2	4825; 6910	230
10965	1	4825; 5600	540
12965	3	4825; 5600	2540
12965	2	7530; 4825	610
11965	2	9450	2515
11965	2	8480	3485

The effectiveness of the S-CUT is difficult to assess because so far no similar algorithm has been described in the literature. It can be assumed that some companies facing such problems do not use any special method for solving them or their methods are very simple. Therefore, the proposed method is first compared with the case where no special method for suborders is used. Suborders are formed randomly (SFR). In the second case, a very simple method was created where orders are first sorted by decreasing order lengths. The obtained list is then divided into two equal parts. The first part contains larger and the second shorter order lengths. Suborders are then formed by picking pairs of corresponding order lengths from both parts, for instance the first one from the first part and the first one from the second, the second one from the first part and the second one from the second and so on. The described simple algorithm is called SA (simple algorithm). Two elements that reduce the trim-loss are taken into account in the SA: there is a greater variety of order lengths in suborders and if suborders are processed in the same order as they were created, then longer order lengths are processed first.

The S-CUT algorithm is tested by solving 108 problem instances randomly generated with the problem generator CUTGEN1 [15]. The lower and upper bounds of stock lengths are [6000, 15000] and of order lengths [900, 9000]. The other parameters for problem generation are presented in Tab. 6. The constraints P , N , M and D are the same as in the practical example. The total length of all ordered pieces in each problem instance amounts to around 50% of the material in stock.

All 108 generated problem instances are solved by SFR, SA, S-CUT and without suborders (WS). The results are shown in Fig. 2.

Table 6 Parameters for the generation of problem instances

I	II	III	IV	V
1-27	17	00000001-00000027	6; 20 7; 20 8; 20	9 9 9
28-54	18	00000028-00000054	9; 20 10; 20 11; 20 12; 20 13; 20 14; 20 15; 20 16; 20	3 4 3 4 3 4 3 3
55-81	19	00000055-00000081	17; 20 18; 20 19; 20 20; 20 21; 20 22; 20 23; 20 24; 20	3 4 3 4 3 4 3 3
82-108	20	00000082-00000108	25; 15 26; 15 27; 15	9 9 9

I - Problem instance number; II - Number of different stock lengths; III - Problem generator seed: from - to; IV - Number of different order lengths (average number of items per length); V - Number of order instances generated.

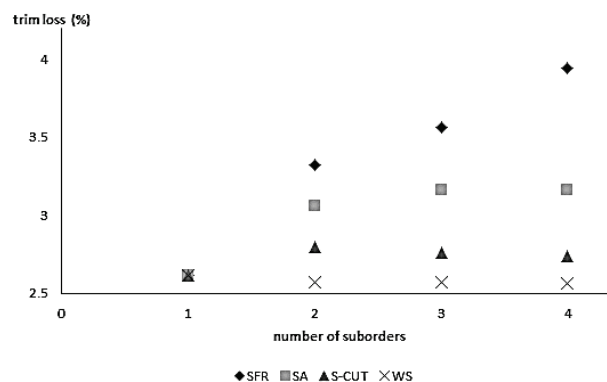


Figure 2 Trim-loss in dependence on the number of suborders

As can be expected, the most efficient method is WS followed by S-CUT, SA and SFR. SFR underperforms S-CUT by 0.78% on average and SA by 0.36% as a percentage of the whole order. S-CUT performs on average 0.22% worse than WS.

Since S-CUT cannot be better than WS, it can be seen that not much room is left for improvement especially because S-CUT can be used with any 1DCSPUL method.

The processing time of an individual problem instance consists of two parts: the time needed for creating suborders and the time for solving suborders. To process randomly generated problem instances on average personal computer, the first part takes on average 10 seconds and the second part less than one.

5 CONCLUSIONS

The proposed heuristic method S-CUT for solving the 1DCSPUL described in this paper splits a large-scale order into smaller suborders. These suborders are then filled independently. S-CUT is used in cases where the entire order cannot be filled in one piece due to various business process limitations. The main objective of S-CUT is the trim-loss minimisation of the entire order. S-CUT takes constraints of a specific practical situation into account.

But very similar constraints can also be expected in other environments. Therefore, S-CUT can be used unchanged or adapted in a variety of companies encountering a similar problem.

The division of a large order into a number of smaller ones is in line with general trends in some industries as a result of the introduction of e-business. One such example is the just-in-time concept. It can therefore be expected that many similar problems will appear in the future.

For further research on cutting stock optimisation, it would be useful if the minimisation of trim-loss were replaced by the minimisation of costs, where the focus would not only be on the cutting activity but also on other related activities that make up the processes of procuring material, storage, logistics and production.

6 REFERENCES

- [1] Madić, M., Radovanović, M., Nedić, B., & Marušić, V. (2015). Multi-objective optimization of cut quality characteristic in CO₂ laser cutting stainless steel. *Technical Gazette*, 22(4), 885-892. <https://doi.org/10.17559/TV-20140211234150>
- [2] Gilmore, P. C. & Gomory, R. E. (1963). A linear programming approach to the cutting stock problem – Part II. *Operations Research*, 11(6), 863-888. <https://doi.org/10.1287/opre.11.6.863>
- [3] Cui, Y., Gu, T., & Hu, W. (2009). A cutting-and-inventory control problem in the manufacturing industry of stainless steel wares. *Omega*, 37(4), 864-875. <https://doi.org/10.1016/j.omega.2008.05.005>
- [4] Chauhan, S. S., Martel, A., & D'amour, S. (2008). Roll assortment optimization in a paper mill: An integer programming approach. *Computers & Operations Research*, 35(2), 614-627. <https://doi.org/10.1016/j.cor.2006.03.026>
- [5] Venkateswarlu, P. (2001). The trim loss problem in wooden container manufacturing company. *Journal of Manufacturing Systems*, 20(3), 166-176. [https://doi.org/10.1016/S0278-6125\(01\)80038-8](https://doi.org/10.1016/S0278-6125(01)80038-8)
- [6] Wang, S., Liang, Z., Wang, B., Zhang, C., & Rahman, Z. (2007). Precise cutting of single-walled carbon nanotubes. *Nanotechnology*, 18(5), 1-7. <https://doi.org/10.1088/0957-4484/18/5/055301>
- [7] Gradišar, M., Resinovič, G., Jesenko, J., Kljajić, M. (1999). A sequential heuristic procedure for one-dimensional cutting. *European Journal of Operational Research*, 114(3), 557-568. [https://doi.org/10.1016/S0377-2217\(98\)00140-4](https://doi.org/10.1016/S0377-2217(98)00140-4)
- [8] Jin, M., Ge, P., & Ren, P. (2015). A new heuristic algorithm for two-dimensional defective stock guillotine cutting stock problem with multiple stock sizes. *Technical Gazette*, 22(5), 1107-1116. <https://doi.org/10.17559/TV-20150731113849>
- [9] Alves, C. & de Carvalho, J. M. V. (2008). New integer programming formulations and an exact algorithm for the ordered cutting stock problem. *Journal of the Operational Research Society*, 59(11), 1520-1531. <https://doi.org/10.1057/palgrave.jors.2602494>
- [10] Kharoufeh, J. P., Cox, S. M., & Oxley, M. E. (2013). Reliability of manufacturing equipment in complex environments. *Annals of the operations research*, 209(1), 231-254. <https://doi.org/10.1007/s10479-011-0839-x>
- [11] Erjavec, J., Trkman, P., & Gradišar, M. (2008). Renovation of the cutting stock process. *International Journal of Production Research*, 47(14), 3979-3996. <https://doi.org/10.1080/00207540801935624>
- [12] Li, S. (1996). Multi-job cutting stock problem with due dates and release dates. *The Journal of the Operational Research Society*, 47(4), 490-510. <https://doi.org/10.1057/jors.1996.56>
- [13] Trkman, P. & Gradišar, M. (2007). One-dimensional cutting stock optimization in consecutive time periods. *European Journal of Operational Research*, 179(2), 291-301. <https://doi.org/10.1016/j.ejor.2006.03.027>
- [14] Alem, D. J., Muniari, P. A., Arenales, M. N., & Ferreira, P. A. V. (2010). On the cutting stock problem with the stochastic demand. *Annals of the Operations Research*, 179(1), 169-186. <https://doi.org/10.1007/s10479-008-0454-7>
- [15] Gau, T. & Wäscher, G. (1995). CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem. *European Journal of Operational Research*, 84(3), 572-579. [https://doi.org/10.1016/0377-2217\(95\)00023-J](https://doi.org/10.1016/0377-2217(95)00023-J)

Contact information:

dr. Mirko GRADIŠAR, Full Professor
University of Ljubljana, Faculty of Economics,
Kardeljeva Ploščad 17, 1000 Ljubljana, Slovenia
miro.gradisar@ef.uni-lj.si

dr. Mihael CESAR, Junior Researcher
University of Ljubljana, Faculty of Economics,
Kardeljeva Ploščad 17, 1000 Ljubljana, Slovenia
mihael.cesar@bsi.si

dr. Luka TOMAT, Researcher and Teaching Assistant
University of Ljubljana, Faculty of Economics,
Kardeljeva Ploščad 17, 1000 Ljubljana, Slovenia
luka.tomat@ef.uni-lj.si