International Journal of Applied Industrial Engineering
Volume 6 • Issue 2 • July-December 2019

A Least-Loss Algorithm for a Bi-Objective One-Dimensional Cutting-Stock Problem

Hesham K. Alfares, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
https://orcid.org/0000-0003-4040-2787

Omar G. Alsawafy, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

ABSTRACT

This article presents a new model and an efficient solution algorithm for a bi-objective one-dimensional cutting-stock problem. In the cutting-stock—or trim-loss—problem, customer orders of different smaller item sizes are satisfied by cutting a number of larger standard-size objects. After cutting larger objects to satisfy orders for smaller items, the remaining parts are considered as useless or wasted material, which is called "trim-loss." The two objectives of the proposed model, in the order of priority, are to minimize the total trim loss, and the number of partially cut large objects. To produce near-optimum solutions, a two-stage least-loss algorithm (LLA) is used to determine the combinations of small item sizes that minimize the trim loss quantity. Solving a real-life industrial problem as well as several benchmark problems from the literature, the algorithm demonstrated considerable effectiveness in terms of both objectives, in addition to high computational efficiency.

KEYWORDS

Cutting-Stock Problem, Heuristic Algorithms, Multiple-Objective Optimization, Trim-Loss Problem

1. INTRODUCTION

The trim-loss or cutting stock problem (CSP) is an important applied optimization problem. CSP assumes a given a number of standard sizes of large objects, and customer demands for different quantities of smaller pieces. A CSP solution specifies the number of smaller pieces cut from each large standard-size object. Of course,

DOI: 10.4018/IJAIE.2019070101

many smaller-piece cut combinations may not consume the full size of the larger objects, resulting in smaller, unused remainders called trim loss. The main objective of CSP is to minimize the total trim loss (wasted material) left over after cutting all larger objects necessary to satisfy customer orders. In general, optimum solutions are difficult for practical, industrial-size CSP problems. According to Garey and Johnson (1979), CSP is a complex, NP-complete optimization problem. Therefore, heuristic techniques are usually used to solve real-world, applied trim-loss problems.

Cutting stock problems (CSP) are classified according to several criteria, but mainly according to the dimension of the problem. One-dimensional problems (1D-CSP) involve one-dimensional (i.e., length) cut decisions, as in the cutting of paper, fabric, and cable rolls that have the same width. Two-dimensional problems (2D-CSP) involve two-dimensional (i.e., length and width) cut decisions, as in the cutting of wood and glass. Three-dimensional problems (3D-CSP) have very limited applications in industry and in the literature. However, there are few recent exceptions, such as the 3-D knapsack problem considered by Baldi et al. (2012). This paper is concerned with a bi-objective one-dimensional cutting stock problem (1D-CSP).

Dyckhoff (1990) presents a four-criterion typology of cutting and packing problems, considering them as two types of "geometric combinatorics" problems related by material-space duality. Wäscher et al. (2007) enhance Dyckhoff's classification and propose a five-criterion typology. According to the four-criterion topology of Dyckhoff (1990), the 1D-CSP problem considered in this paper can be classified as a classical cutting stock problem denoted by (1/V/I/R). Here, (1) denotes 1 dimension, (V) indicates all small items must be cut from a selection of large objects, (I) denotes identical (single) size of large objects, and (R) denotes many small items of the same shape. However, the classical cutting stock problem has a single objective (minimizing trim loss), while our problem has two objectives.

One-dimensional problems (1D-CSP) models may consider either a single given size or a few given standard sizes for all available large objects. Most previous 1D-CSP models focus on only minimizing the total trim loss quantity. This paper presents a bi-objective 1D-CSP model with one given large-object size, where the primary objective is to minimize the total amount of trim loss. The second objective is to minimize the number of partially cut large objects. Minimizing the number of partially cut large objects is a real-life objective for many companies, because partially cut objects are generally more difficult to reuse or cut again. Moreover, partially cut stocks are not as easy or profitable to sell as uncut objects.

An integer linear programming (ILP) model of the problem is formulated to determine the optimum number of used large objects, and the cutting pattern for each large object. As the optimum solution of this ILP model is difficult to obtain, a two-stage heuristic least-loss algorithm (LLA) is developed to solve the problem effectively and efficiently. In the first stage, a decreasing order of size is used to assign small items to cutting patterns in order to minimize trim loss. If the solution does not satisfy a specific performance criterion, then a second stage is required in which a different initial order of small items is used. The algorithm is effectively applied

to a real-life industrial cutting-stock problem. Moreover, numerical comparisons on benchmark problems are carried out, demonstrating the superior performance of the proposed algorithm.

The paper presents a bi-objective mathematical model and heuristic algorithm to determine the number of large objects used, and the cutting pattern for each object. Subsequent sections of this paper are organized as follows. Relevant literature is surveyed in Section 2. The mathematical optimization model is formulated in Section 3. The two-stage least-loss algorithm (LLA) is described in Section 4. Results of the computational comparisons and the industrial application are presented in Section 5. Finally, conclusions and suggestions are provided in Section 6.

2. LITERATURE REVIEW

Because of its practical value and theoretical significance, the trim-loss cutting-stock problem is one of the most popular and well-studied problems in operations research. Dyckhoff (1990) reviews literature up to 1990, listing 16 previous surveys of literature on various aspects of cutting and packing problems from 1971 to 1988. Wäscher et al. (2007) review literature from 1995 to 2004. Recently, Melega et al. (2018) review and classify the literature relevant to the integration between the cutting stock problem and the production lot-sizing problem. In this section, the primary concern is one-dimensional cutting stock problem (1D-CSP) literature published since 2005. Because this paper presents a new bi-objective model, more emphasis is given to recent multi-objective 1D-CSP techniques.

Some 1D-CSP models assumes that the trim loss is not completely lost, but may have some salvage value. Cherri et al. (2013) develop a heuristic method for a multi-period 1D-CSP with usable leftover, assuming the trim material is reusable if large enough, but it has to be consumed quickly. Over a specific time horizon, new demands in each period are satisfied from left-over and remaining unused large-objects. Cui et al. (2017) consider a 1D-CSP where specific types of leftover can be used to fulfill new orders. A heuristic algorithm is developed based on column generation procedure to solve such problem. Garraffa et al. (2016) present a pattern-based heuristic to solve the 1D-CSP with sequence-dependent cutting losses due to cutting processes by different tools. The main objective is to minimize cutting patterns, and the secondary objective is to have larger left-overs in order to maximize their usability.

Many techniques in the literature deal with bi-objective 1D-CSP. A pattern-free bi-objective ILP model is formulated by Kasimbeyli et al. (2011) for 1D-CSP. The first objective is to minimize total trim loss, while the second is to minimize the total number of rolls (large objects) used. A heuristic algorithm is used to solve this ILP model efficiently. Aliano Filho et al. (2018) optimize two objectives for 1D-CSP: minimizing the number of cutting patterns, and minimizing the frequency of their use. Four solution techniques are compared: weighted sum, Chebyshev metric, ε-Constraint, and the modified Chebyshev metric. The two objectives of minimum trim loss (material cost) and minimum cutting patterns (setup cost) are combined by

Mobasher and Ekici (2013) into one total production cost. A mixed integer linear programming model is formulated, and three heuristic solution methods are developed based on local search and column generation concepts. Ogunranti and Oluleye (2016) investigate the 1D-CSP in wood industry. A pattern generation algorithm is integrated with a linear programming (LP) model to minimize trim loss and the number of used stocks. In addition to minimizing the trim loss, Arbib et al. (2016) assume that cutting patterns must be sequenced to satisfy a given limit on the number of different part types in production at any time. The problem is formulated as ILP model, and a column generation procedure is used to solve it.

Multiple-objective 1D-CSP models aim to achieve several simultaneous, often conflicting, objectives. Matsumoto et al. (2011) consider a multi-objective 1D-CSP in the paper industry to minimize four objectives. The problem is solved by generating cutting pattern sequences, decomposing into several bin-packing problems, and tabu search. Cui et al. (2015) solve a 1D-CSP problem with two objectives, minimizing the material cut cost and the number of different patterns used (setup cost). First a set of patterns is generated using sequential grouping procedure, and then this set is used in solving the ILP model. A dynamic programming (DP)-based heuristic is used by Tanir et al. (2016) to solve a 1D-CSP problem with divisible items, where some of the leftovers can be combined by welding to form demanded items. The objective is to minimize both the trim-loss material and the number of the welding operations. Son et al. (2016) study the cutting process in window frame manufacturing using mixed integer programming and a knapsack-based heuristic approach. The aim is to minimize the weighted sum of trim loss, bar type imbalance, and the degree of order spreading. Liu et al. (2017) consider a cutting-stock problem to minimize three objectives: trim loss, number of cutting patterns, and usable leftovers. Multi-objective optimization, heuristic methods, and multi-attribute decision-making are combined to solve this problem.

Although the 1D-CSP problem is NP-complete, optimal solutions have been developed for special cases taking advantage of the unique problem structure. An optimization model and solution technique are presented by Reinertsen and Vossen (2010) for 1D-CSP with due date constraints. The model is applied to an industrial case study, incorporating realistic considerations such as order aggregation, multiple stock lengths, and rolling horizons. A model is developed by Alves and de Carvalho (2008) to minimize the number of setups in the cutting-stock problem. To solve the NP-hard problem, a branch-and-price-and-cut algorithm is used to generate stronger cuts. The algorithm utilizes dual feasible functions, valid inequalities, and arc flow variables to strengthen the bounds on the column generation model. A similar cutting stock problem is considered by Arana-Jiménez and Neto (2017) with two objectives: minimizing the number of used large objects, and minimizing the number of cutting patterns. Integrating discrete optimization with continuous vector optimization, sufficient conditions for partial optimality are derived for this problem. Muter and Sezer (2018) solve the two-stage 1D-CSP by designing an exact simultaneous column-and-row generation algorithm.

Several meta-heuristic techniques have been applied for the trim-loss problem, and specifically for 1D-CSP. These techniques include simulated annealing, tabu search, and genetic algorithms. Two meta-heuristic algorithms, simulated annealing (SA) and tabu search (TS), are applied to the 1D-CSP by Jahromi et al. (2012). Comparing the results on several test problems, it is concluded that SA outperforms TS in terms of the objective function values. A genetic algorithm-based heuristic is proposed to solve a bi-objective 1D-CSP by Araujo et al. (2014). The two objectives are minimizing the number of used large objects and minimizing the number of different cutting patterns. In addition, as discussed earlier, tabu search is used by Matsumoto et al. (2011). Evtimov and Fidanova (2018) use ant-colony optimization algorithm to solve a cutting-stock problem. Sanchez et al. (2018) solve the binary cutting stock problem using ILP and three metaheuristic algorithms: genetic algorithms, simulated annealing, and particle swarm optimization.

A number of papers combine the cutting-stock problem with other related problems. Arbib and Marinelli (2014) combine the classical cutting-stock problem with the machine scheduling problem by considering different due dates for orders. Poldi and de Araujo (2016) study the combined problem of cutting stock and production planning, considering a finite planning horizon and taking the cost of inventory into consideration. Melega et al. (2016) develop a model for the integrated lot-sizing and cutting-stock problem, and then solve it using a commercial optimization package and a column generation technique. Vanzela et al. (2017) consider the integrated cutting-stock and lot-sizing problem in small-scale furniture factories. Constraints related to the cutting saw are considered as limits on the cutting capacity. Poltroniere et al. (2016) integrate the lot-sizing problem with the cutting stock problem in the paper industry and solve the integrated model both heuristically and optimally. Leao et al. (2017) discuss a similar problem in the paper industry, in which 1D-CSP is integrated with the lot-sizing problem. Column generation techniques and rounding heuristics are developed to obtain feasible solutions.

The above literature review indicates that multi-objective 1D-CSP approaches are very common in the literature. However, with the exception of Liang et al. (2002), none of the published papers simultaneously addresses the same two specific objectives of this paper. As far the authors know, only Liang et al. address the same two objectives: minimizing the trim loss, and minimizing the number of partially cut large objects. Considering these two objectives, Liang et al. (2002) apply an evolutionary programming (EP) heuristic to 1D-CSP with and without contiguity. Computational comparisons using 20 test problems demonstrate that their EP solutions are significantly better than genetic algorithms (GA) solutions in most cases and equivalent in the remaining cases. In this paper, a heuristic two-stage least-loss algorithm (LLA) is developed for the bi-objective 1D-CSP and it is compared with the EP heuristic of Liang et al. (2002). This paper is an extended and enhanced version of Alfares and Alsawafy (2017) conference paper.

3. PROBLEM DEFINITION AND MODEL FORMULATION

To be consistent with the terminology of Dyckhoff (1990), it is assumed that an unlimited number of "*large objects*" of one standard size need to be cut into specified quantities and sizes of "*small items*". A "*cutting pattern*" indicates the number of times each small item size is cut from a *single* large object. A "*cutting plan*" is a complete solution that specifies cutting pattern frequencies, i.e., the number of large objects cut according to each pattern.

The sizes and quantities of the small items are assumed to be known values specified by customer orders. Orders are classified according to the size of small items, thus orders by different customers for the same size are combined together. The primary objective of our model is to minimize the total trim loss. The second objective is to minimize the number of partially used large objects. According to our experience, partially used large objects are considered as left-over stock that may or may not be used again. For the company in which the real-life problem is solved, partially cut objects are more difficult to use and less profitable to sell. The ILP model of the 1D-CSP problem described above is presented below. First, the notation is defined, and then the model objectives and constraints are presented.

3.1. Integer Programming Model Notation

s_i = size (constant length) of all small items in order i, $i = 1, \ldots, I$
q_i = quantity of small items of size s_i required in order i, $i = 1, \ldots, I$, $q_i \geq 0$ and integer
a_{ij} = number of times size s_i is cut from cutting pattern j, $i = 1, \ldots, I$, $j = 1, \ldots, J$, $a_{ij} \geq 0$ and integer
w_j = trim loss of cutting pattern j, $0 \leq w_j \leq L$, $j = 1, \ldots, J$
b_j = 1 if cutting pattern j has trim loss ($w_j > 0$), $j = 1, \ldots, J = 0$ if cutting pattern j has no trim loss ($w_j = 0$)
L = given standard length of the large objects, $L \geq 0$
N = number of large objects used, $N \geq 0$ and integer
PC = number of partially-cut large objects ($PC \leq N$), $PC \geq 0$ and integer
X_j = number of times cutting pattern j is used, i.e., number of large objects cut according to cutting pattern j, $X_j \geq 0$ and integer, $j = 1, \ldots, J$

3.2. Objective Functions

The first objective (1) is to minimize the total trim loss TL, and the second objective (2) is to minimize the number of partially cut large objects PC:

$$\text{Minimize } TL = \sum_{j=1}^{J} w_j X_j \tag{1}$$

$$\text{Minimize } PC = \sum_{j=1}^{J} b_j X_j \tag{2}$$

3.3. Constraints

Customer demands for each order (size) must be satisfied:

$$\sum_{j=1}^{J} a_{ij} X_j \geq q_i \; i = 1, \ldots, I \tag{3}$$

The number of used large objects is equal to the total frequency of all cutting patterns:

$$\sum_{j=1}^{J} X_j = N \tag{4}$$

The trim loss of cutting pattern j is the difference in lengths between the large object and all small items cut from it:

$$w_j = L - \sum_{i=1}^{I} a_{ij} s_i \; j = 1, \ldots, J \tag{5}$$

$$L.b_j \geq w_j j = 1, \ldots, J \tag{6}$$

Although the above ILP model looks fairly simple, it is quite difficult to construct and to solve optimally due to several factors. First, the generation of cutting patterns alone is a challenging task. For a given large object length L, the number of possible patterns increases exponentially with the number of orders (sizes) I. Second, the inclusion of two objectives in the above model is a significant complicating factor that makes optimum solution more difficult. Handling the two objectives by pre-emptive goal programming would require two stages of optimum solution in the order of two priorities. Third, the pure integer optimization model is difficult to solve for large-size industrial problems. Finally, the nonlinearity of the second objective function (2) adds another dimension of difficulty, as nonlinear optimization problems are generally far more formidable than linear problems. Based on the difficulty in obtaining the optimum solution, heuristic methods constitute the only practical option to solve the problem. In this paper, logical heuristic rules are used to develop a two-stage least-loss solution algorithm, which is presented below.

4. TWO-STAGE LEAST-LOSS ALGORITHM (LLA)

4.1. Algorithm Notation

D = allowed trim loss for the given iteration
RM = remainder (actual trim loss) for the given pattern
k = number of small items in each large object
MX_i = maximum number of small items in each large object if size s_i is included
S = vector of required small item sizes = $(s_1, s_2, \ldots, s_n)$

Q = vector of the quantity (number required) of small item sizes = $(q_1, q_2, \ldots, q_n)$
$\delta_{k,i}$ = one pattern (combination) of k small items including s_i, $\delta_{k,i} \in \nabla(k,i)$
$\nabla(k,i)$ = set of all different patterns $\delta_{k,i}$ containing k small items including s_i, such that all sizes are not less than s_i, and the sum of their sizes is not greater than L
$PF(\delta_{k,i})$ = pattern frequency, i.e., number of times pattern $\delta_{k,i}$ is used
$ns_{j\delta}$ = number of times size s_j is used (cut) in pattern $\delta_{k,i}$

4.2. Stage 1: Decreasing Order of Size

In order to solve the problem, the heuristic least-loss algorithm (LLA) proceeds in two stages. In the first stage, the small items are arranged in decreasing order of size. This decreasing order has been found to provide the best start for the algorithm, as it generally leads to the best heuristic solutions. Due to the effectiveness of the decreasing size order, it is used in several widely-used CSP solution methods, including the well-known first-fit-decreasing (FFD) heuristic (Eilon and Christofides, 1971). The FFD algorithm arranges small items in decreasing order of length, and then assigns them individually to the first available large object. According to Dyckhoff (1990), the FFD algorithm is quite efficient, having a time complexity of $O(I \log I)$ and a worst-case performance of 18.2% trim-loss increase above the optimum.

During the first stage, the least-loss algorithm proceeds in decreasing order of small item sizes ($s_1 > s_2 > s_3 > \ldots > s_n$). Starting with size 1 (largest item), the algorithm first finds all possible combinations of the current size (i) and smaller sizes (s_i, …, s_n) that produce zero trim-loss ($D = 0$). After assigning all available small items to these combinations, the remaining sizes and quantities are determined. Next, the algorithm moves to the next (smaller) item size to find and assign all combinations of the remaining small items sizes for which $D = 0$. After going through all sizes, the algorithm starts again at size 1 to finds all possible combinations of the remaining items that produce a one-unit loss ($D = 1$). The process is repeated until all small items have been assigned to large items, i.e., all customer orders have been satisfied. Steps of the least-loss algorithm are depicted in Figure 1, and details of these steps are described below.

Initialization step:

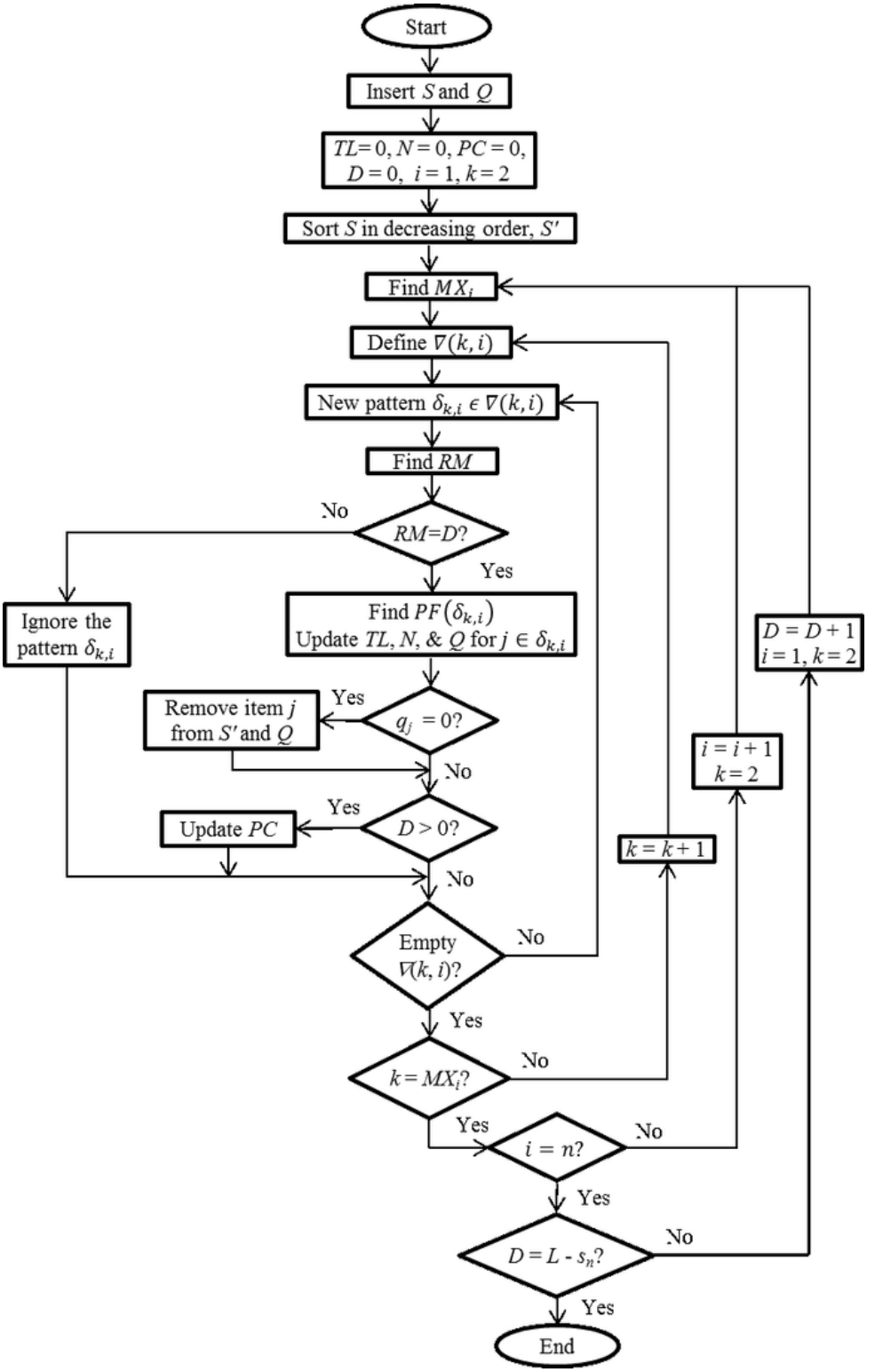
Set $TL = 0$
Set $N = 0$
Set $PC = 0$

4.2.1. Step 1: Initial Order

Arrange small items in decreasing order of size:

$s_1 > s_2 > s_3 > \ldots > s_n$

Figure 1. Flowchart of the least-loss algorithm



4.2.2. Step 2: Least-Loss Assignments

For $D = 0, \ldots, L - s_n$
  For $i = 1, \ldots, n$

$$MX_i = \left\lfloor \frac{L - s_i}{s_n} \right\rfloor + 1,$$

Where $\lfloor a \rfloor$ = largest integer $\leq a$

For $k = 2, \ldots, MX_i$

For all $\delta_{k,i} \in \nabla(k,i)$

$$RM = L - \sum_{j \in \delta_{k,i}} s_j$$

If $RM = D$

Select pattern $\delta_{k,i}$

$$PF(\delta_{k,i}) = \min_{j \in \delta_{k,i}} \left\{ \left\lfloor \frac{q_j}{n s_{j\delta}} \right\rfloor \right\}$$

$q_j = q_j - n s_{j\delta} \times PF(\delta_{k,i})$, $j \in \delta_{k,i}$

If $q_j = 0$, remove item j from arrays S and Q

$TL = TL + D \times PF(\delta_{k,i})$

$N = N + PF(\delta_{k,i})$

If $D \neq 0$, $PC = PC + PF(\delta_{k,i})$

If $RM \neq D$

Ignore pattern $\delta_{k,i}$ and go to the next pattern

End if

End for $\delta_{k,i}$

End for k

End for i

End for D

If the first stage of the LLA heuristic produces a satisfactory solution, then a second stage is not required. In order to consider the first-stage solution satisfactory, the number of large objects used, N, should be no more than 5% percent above the optimum lower bound $N_{\min}$. This 5%-threshold is a heuristic parameter that has been determined based on extensive numerical experiments.

If:

$$\frac{100(N - N_{\min})}{N_{\min}} > 5\%$$

go to Phase 2, where:

$$N_{\min} = \left\lceil \frac{1}{L} \sum_{i=1}^{I} q_i s_j \right\rceil \tag{7}$$

If this condition is not satisfied, then the algorithm proceeds to the second stage to try to obtain a better solution.

4.3. Stage 2: Modified Initial Order

The second stage of the algorithm is similar to the first, but it starts with a different order of small items. After arranging small items in decreasing order of size, the middle-ranked item is moved to the beginning of the sequence. This second-stage rearrangement of small items is selected from several reordering options on the basis of extensive numerical experimentation.

This modified sequence usually leads to better solutions if the first-stage solution is found unsatisfactory. The second stage is a neighborhood search around the first-stage solution, where the neighborhood move is defined as a permutation of the order of small items. At the end of the second stage, the algorithm selects the better of the two solutions produced in stages 1 and 2.

4.3.1. Step 1: Initial Order

Arrange small items in decreasing order of size, and then bring the item in the middle to the beginning of the sequence. If the number of sizes is even, take the second item from the two sizes in the middle. As usual with all heuristic procedures, this rule was developed based on extensive trial-and-error and numerical experimentation:

$$s_{\lceil n+1/2 \rceil} > s_1 > s_2 > \ldots > s_{\lceil n+1/2 \rceil -1} > s_{\lceil n+1/2 \rceil +1} > \ldots > s_n$$

4.3.2. Step 2: Least-Loss Assignments

Same steps as in Stage 1.

5. BENCHMARKING AND INDUSTRIAL APPLICATION

5.1. Benchmarking Analysis

In order to test the least-loss algorithm (LLA), we searched for benchmark CSP problems described in the literature. In spite of the abundance of published studies reporting computational experiments with numerous CSP test problems, very few papers provide full descriptions of the test problems. As an exception, Liang et al. (2002) provide complete descriptions of 20 test problems they used in their computational experiments. Out of these 20 problems, we used 10 problems that have a single large-object length. This set of 10 problems contains 5 smaller problems (1-5) originally described by Hinterding and Khan (1995), in which the number of sizes range is (I = 8-18) and the number of items range is (Σq_i = 20-126). The remaining 5 problems (6-10) are larger, in which the number of sizes range is (I = 18-36) and the number of items range is (Σq_i = 200-600). Size dimensions of the 10 benchmark problems are shown in Table 1.

Table 1. Size dimensions and solution times of benchmark problems

Problem Number	Number of Sizes: I	No. of Items: Σq_i	Large Object Length: L	LLA Solution Time (sec)
1	8	20	14	0.125
2	8	50	15	0.1094
3	8	60	25	0.1406
4	8	60	25	0.2969
5	18	126	4300	1.8906
6	18	200	86	0.9219
7	24	200	120	5.0781
8	24	400	120	6.125
9	36	400	120	46.813
10	36	600	120	52.688

Computational experiments with the 10 benchmark problems were performed to evaluate the proposed two-stage least-loss algorithm (LLA) in term of two criteria. The first criterion is efficiency (computational performance, or solution time), while the second criterion is effectiveness (i.e. solution quality, or values of the objective functions). To assess computational efficiency, solution times of the least-loss heuristic are shown in Table 1. These times were obtained using MATLAB 2011a, or version 7.12.0 I, on a laptop PC running on Windows 8, with the following hardware specifications (Intel(R) Core(TM) I7 CPU, 2.20 GHz, 8.00 GB RAM). Unfortunately, comparative and up-to-date computational time data for these problems is not available. However, the solution times are clearly very reasonable, given the large dimensions of the test problems. Even the largest problems, involving 36 sizes and 600 ordered small items, are solved in less than 1 minute, indicating a high level of computational efficiency.

In terms of the solution quality, the proposed least-loss heuristic is compared to both the optimum solution and the Liang et al. (2002) solution. The optimal solutions of the 10 test problems are not available, but we used the theoretical bound $N_{\min}$ to represent the optimal number of large objects, N. Table 2 shows the values of N obtained by three methods: the optimum bound ($N_{\min}$), Liang et al. procedure, and the least-loss algorithm (LLA). Judging by the values of N, the least-loss algorithm produces solutions that are either optimum or very nearly optimum. The least-loss algorithm solution is optimum for 5 test problems and it increases N by an average of only 1.16% above optimum. On the other hand, the solution procedure of Liang et al. (2002) is optimum for 4 test problems and increases N on average by 2.49% above optimum. It should be noted that the LLA solutions might be actually even closer to optimality, because the optimum solution might be in some cases greater than the lower bound $N_{\min}$.

Table 2. Number of large objects used *N* for benchmark problems

Problem	N_{min}	Liang	LLA	% Liang > N_{min}	% LLA > N_{min}
1	9	9	9	0	0
2	23	23	23	0	0
3	15	15	15	0	0
4	19	19	19	0	0
5	51	54	53	5.88	3.92
6	78	82	81	5.13	3.85
7	68	69	68	1.47	0
8	143	149	145	4.20	1.40
9	149	155	152	4.03	2.01
10	215	224	216	4.19	0.47

Table 3 compares the solution quality of the least-loss algorithm (LLA) to the EP heuristic of Liang et al. (2002) in terms of two objectives: trim-loss quantity *TL*, and number of partially-cut large objects *PC*. For the trim-loss objective *TL*, the LLA heuristic produced better solutions for all 10 test problems, reducing *TL* by an average of 26.4%, and a maximum of 87.5% compared to the solutions of Liang et al. (2002). For the number of partially cut large objects *PC*, the LLA heuristic produced better solutions in 9 out of 10 problems, reducing *PC* by an average of 30.5% and a maximum of 85.9% compared to Liang et al. solutions. In general, the least-loss algorithm (LLA) seems to provide a greater advantage of over the Liang et al. solutions as the problem size increases.

In summary, comparisons with the solutions by Liang et al. (2002) confirmed the superiority of the proposed LLA heuristic in term of the solution quality and

Table 3. Trim-loss *TL* and partially-cut objects *PC* for benchmark problems

Problem Number	Liang	*TL* LLA	% LLA < Liang	Liang	*PC* LLA	% LLA < Liang
1	3	3	0	2	2	0
2	13	13	0	4	4	0
3	0	0	0	0	0	0
4	11	11	0	1.2	1	16.67
5	11966	11450	4.31	22.8	23	- 0.88
6	309.4	275	11.12	33.7	28	16.91
7	189.6	84	55.70	15.34	4	73.92
8	788	332	57.87	76.68	32	58.27
9	730	382	47.67	48.4	22	54.55
10	1037.2	130	87.47	70.7	10	85.86

computational performance. The results show significant improvements over the Liang et al. (2002) algorithm, with greater improvements for larger sizes problems. The results also show that the least-loss algorithm provides near-optimum solutions in short computation times.

5.2. Industrial Application

Having proven its comparative superiority, the LLA heuristic was then applied to a real-life large trim-loss problem. The company facing this trim-loss problem is a leading manufacturer of office furniture. The company needs to cut 6m-long bars of extrusion profiles used to make office partitions in order to satisfy demands from different customers. To solve such problems, the company has been using a commercial trim-loss software package that applies an undisclosed (black-box) heuristic. For a given planning period, data was obtained on actual typical demands for different small sizes of extrusion profiles from three branches of the company. This data represents a large-size trim-loss problem, in which the number of ordered items Σq_i is equal to 7,764. This problem was solved by both the LLA heuristic and the company's software package.

Table 4 compares the LLA heuristic solution with the results produced by the commercial software package. Using (7), the theoretical lower bound on the minimum number of large objects required N_{min} is equal to 1454. In the LLA heuristic solution, the number of objects used is 1473, which is only 1.3% above N_{min}. This proves that the LLA heuristic is able to produce near-optimum solutions for large-size industrial problems. Moreover, compared to the company's software package, the LLA heuristic reduces the total trim loss by 14.2% and the number of used large objects by 4.6%. For both the LLA solution and the company's solution, it should be noted, all the used large objects are partially cut, i.e. $N = PC$. Reductions in the total trim loss lead to hundreds of thousands of savings in material costs each year. Moreover, reductions in the number of used large objects result in additional savings in labor hours, material handling, and storage costs.

5.3. Managerial Insights

The results of both the benchmark study and the industrial case application confirm that the proposed algorithm significantly outperforms previous techniques published in the literature as well as an existing commercial software. In particular, the industrial application presented in this section shows that our least-loss algorithm (LLA)

Table 4. Trim-loss and number of large objects for the industrial application

	***TL* (mm)**	***N***	***PC***	**$\% N > N_{min}$**
Co.	570,542	1544	1544	6.2
LLA	489,457	1473	1473	1.3
% LLA < Co.	14.2	4.6	4.6	4.9

heuristic is valid and very beneficial in large-scale real-life industrial applications. For production managers dealing with one-dimensional trim-loss problems, the proposed LLA algorithm provides a viable option to make near-optimum daily decisions. Obviously, these managers must aim to achieve the two prioritized objectives of the LLA heuristic: minimum trim-loss, and minimum number of partially-cut large objectives. If this is the case, then the LLA heuristic has the potential to provide huge savings in terms of the material, labor, and inventory costs. Lowering the number of partially-cut large objects also reduces the processing and operational costs, simplifies planning, and reduces managerial overhead.

6. CONCLUSION

A new, bi-objective, one-dimensional cutting-stock problem (1D-CSP) has been modeled and solved. The two objectives, in the order of priority, are minimization of trim-loss quantity, and minimization of the number of partially cut large objects. Assuming that a single standard length is specified for all large objects, the integer-programming model of this multi-objective 1D-CSP has been formulated. As the optimum solution of this problem is not easy to find, especially for large industrial applications, a new heuristic least-loss algorithm (LLA) has been presented to efficiently produce near-optimum solutions. Based on comparative experiments with benchmark problems, the new LLA heuristic demonstrated significant advantage over previous approaches in the literature. Computational tests also confirmed the near-optimality and computational efficiency of this two-stage heuristic algorithm. The significant reductions obtained by the proposed algorithm in trim-loss and number of partially cut large objects mean huge savings in material, processing, and inventory costs.

Several alternatives to extend this work are possible for future research. These include the consideration of several standard lengths of large objects instead of only one standard length. Another possibility is to apply a modified least-loss approach to higher-dimensionality problems such as 2-D CSP and 3-D CSP. An additional interesting extension would be to include explicit cost objectives such as setup cost and inventory cost. Many other logical extensions are available, including considering multiple time periods, shortages, and time-varying or stochastic customer demands for the small items.

REFERENCES

Alfares, H. K., & Alsawafy, O. G. (2017). Efficient least-loss algorithm for a bi-objective trim-loss problem. In *Proceedings of the 2017 International Conference on Industrial Engineering and Operations Management*, Rabat, Morocco (pp. 2118-2122). Academic Press.

Aliano Filho, A., Moretti, A. C., & Pato, M. V. (2018). A comparative study of exact methods for the bi-objective integer one-dimensional cutting stock problem. *The Journal of the Operational Research Society*, *69*(1), 91–107. doi:10.1057/s41274-017-0214-7

Alves, C., & de Carvalho, J. V. (2008). A branch-and-price-and-cut algorithm for the pattern minimization problem. *Operations Research*, *42*(4), 435–453. doi:10.1051/ro:2008027

Arana-Jiménez, M., & Neto, L. S. (2017). Sufficient condition for partial efficiency in a bicriteria nonlinear cutting stock problem. *Operations Research*, *51*(3), 709–717. doi:10.1051/ro/2016058

Araujo, S. A. D., Poldi, K. C., & Smith, J. (2014). A genetic algorithm for the one-dimensional cutting stock problem with setups. *Pesquisa Operacional*, *34*(2), 165–187. doi:10.1590/0101-7438.2014.034.02.0165

Arbib, C., & Marinelli, F. (2014). On cutting stock with due dates. *Omega*, *46*, 11–20. doi:10.1016/j.omega.2014.01.004

Arbib, C., Marinelli, F., & Ventura, P. (2016). One-dimensional cutting stock with a limited number of open stacks: Bounds and solutions from a new integer linear programming model. *International Transactions in Operational Research*, *23*(1–2), 47–63. doi:10.1111/itor.12134

Baldi, M. M., Perboli, G., & Tadei, R. (2012). The three-dimensional knapsack problem with balancing constraints. *Applied Mathematics and Computation*, *218*(19), 9802–9818. doi:10.1016/j.amc.2012.03.052

Cherri, A. C., Arenales, M. N., & Yanasse, H. H. (2013). The usable leftover one-dimensional cutting stock problem—a priority-in-use heuristic. *International Transactions in Operational Research*, *20*(2), 189–199. doi:10.1111/j.1475-3995.2012.00868.x

Cui, Y., Cui, Y. P., & Zhao, Z. (2015). Pattern-set generation algorithm for the one-dimensional multiple stock sizes cutting stock problem. *Engineering Optimization*, *47*(9), 1289–1301. doi:10.1080/0305215X.2014.969726

Cui, Y., Song, X., Chen, Y., & Cui, Y. P. (2017). New model and heuristic solution approach for one-dimensional cutting stock problem with usable leftovers. *The Journal of the Operational Research Society*, *68*(3), 269–280. doi:10.1057/s41274-016-0098-y

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, *44*(2), 145–159. doi:10.1016/0377-2217(90)90350-K

Eilon, S., & Christofides, N. (1971). The loading problem. *Management Science*, *17*(5), 259–268. doi:10.1287/mnsc.17.5.259

Evtimov, G., & Fidanova, S. (2018). Ant Colony Optimization Algorithm for 1D Cutting Stock Problem BT - Advanced Computing in Industrial Mathematics. In Proceedings of the *11th Annual Meeting of the Bulgarian Section of SIAM* December 20-22, 2016, Sofia, Bulgaria (pp. 25–31). Cham: Springer International Publishing.

Garey, M. R., & Johnson, D. S. (2002). Computers and intractability (Vol. 29). New York: W.H. Freeman.

Garraffa, M., Salassa, F., Vancroonenburg, W., Vanden Berghe, G., & Wauters, T. (2016). The one-dimensional cutting stock problem with sequence-dependent cut losses. *International Transactions in Operational Research*, *23*(1-2), 5–24. doi:10.1111/itor.12095

Hinterding, R., & Khan, L. (1995). Genetic algorithms for cutting stock problems: with and without contiguity. In Progress in evolutionary computation (pp. 166-186). Springer.

Jahromi, M. H., Tavakkoli-Moghaddam, R., Makui, A., & Shamsi, A. (2012). Solving a one-dimensional cutting stock problem by simulated annealing and tabu search. *Journal of Industrial Engineering International*, *8*(1), 24. doi:10.1186/2251-712X-8-24

Kasimbeyli, N., Sarac, T., & Kasimbeyli, R. (2011). A two-objective mathematical model without cutting patterns for one-dimensional assortment problems. *Journal of Computational and Applied Mathematics*, *235*(16), 4663–4674. doi:10.1016/j.cam.2010.07.019

Leao, A. A. S., Furlan, M. M., & Toledo, F. M. B. (2017). Decomposition methods for the lot-sizing and cutting-stock problems in paper industries. *Applied Mathematical Modelling*, *48*, 250–268. doi:10.1016/j.apm.2017.04.010

Liang, K. H., Yao, X., Newton, C., & Hoffman, D. (2002). A new evolutionary approach to cutting stock problems with and without contiguity. *Computers & Operations Research*, *29*(12), 1641–1659. doi:10.1016/S0305-0548(01)00039-9

Liu, L., Liu, X., Pei, J., Fan, W., & Pardalos, P. M. (2017). A study on decision making of cutting stock with frustum of cone bars. *Operations Research*, *17*(1), 187–204. doi:10.1007/s12351-015-0221-x

Matsumoto, K., Umetani, S., & Nagamochi, H. (2011). On the one-dimensional stock cutting problem in the paper tube industry. *Journal of Scheduling*, *14*(3), 281–290. doi:10.1007/s10951-010-0164-2

Melega, G. M., de Araujo, S. A., & Jans, R. (2016). Comparison of Mip Models for the Integrated Lot-Sizing and One-Dimensional Cutting Stock Problem. *Pesquisa Operacional*, *36*(1), 167–196. doi:10.1590/0101-7438.2016.036.01.0167

Melega, G. M., de Araujo, S. A., & Jans, R. (2018). Classification and literature review of integrated lot-sizing and cutting stock problems. *European Journal of Operational Research*, *271*(1), 1–19. doi:10.1016/j.ejor.2018.01.002

Mobasher, A., & Ekici, A. (2013). Solution approaches for the cutting stock problem with setup cost. *Computers & Operations Research*, *40*(1), 225–235. doi:10.1016/j.cor.2012.06.007

Muter, İ., & Sezer, Z. (2018). Algorithms for the one-dimensional two-stage cutting stock problem. *European Journal of Operational Research*, *271*(1), 20–32. doi:10.1016/j.ejor.2018.04.042

Ogunranti, G. A., & Oluleye, A. E. (2016). Minimizing waste (off-cuts) using cutting stock model: The case of one dimensional cutting stock problem in wood working industry. *Journal of Industrial Engineering and Management*, *9*(3), 834–859. doi:10.3926/jiem.1653

Poldi, K. C., & de Araujo, S. A. (2016). Mathematical models and a heuristic method for the multiperiod one-dimensional cutting stock problem. *Annals of Operations Research*, *238*(1-2), 497–520. doi:10.1007/s10479-015-2103-2

Poltroniere, S. C., Araujo, S. A., & Poldi, K. C. (2016). Optimization of an Integrated Lot Sizing and Cutting Stock Problem in the Paper Industry. *TEMA (São Carlos)*, *17*(3), 305. doi:10.5540/tema.2016.017.03.0305

Reinertsen, H., & Vossen, T. W. (2010). The one-dimensional cutting stock problem with due dates. *European Journal of Operational Research*, *201*(3), 701–711. doi:10.1016/j.ejor.2009.03.042

Sanchez, I. A. L., Vargas, J. M., Santos, C. A., Mendoza, M. G., & Moctezuma, C. J. M. (2018). Solving binary cutting stock with matheuristics using particle swarm optimization and simulated annealing. *Soft Computing*, *22*(18), 6111–6119. doi:10.1007/s00500-017-2666-8

Son, D., Kim, B.-I., Bae, B., Park, J.-S., & Ki, Y. (2016). An algorithm for a cutting problem in window frame production. *International Journal of Production Research*, *54*(14), 4327–4339. doi:10.1080/00207543.2016.1148279

Tanir, D., Ugurlu, O., Guler, A., & Nuriyev, U. (2016). One-dimensional Cutting Stock Problem with Divisible Items.

Vanzela, M., Melega, G. M., Rangel, S., & de Araujo, S. A. (2017). The integrated lot sizing and cutting stock problem with saw cycle constraints applied to furniture production. *Computers & Operations Research*, *79*, 148–160. doi:10.1016/j.cor.2016.10.015

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, *183*(3), 1109–1130. doi:10.1016/j.ejor.2005.12.047

Hesham K. Alfares is Chairman and Professor in the Systems Engineering Department at King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia. He has a PhD in Industrial Engineering from Arizona State University. He has more than 100 publications, including journal papers, conference papers, and book chapters. He has been a member in the editorial boards of three international journals and in the program committees of 29 international conferences.

Omar Alsawafy is currently an Assistant Professor in Systems Engineering Department at King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia. He received his PhD. and MSc. degrees from KFUPM in the same field of production planning and inventory control. His research interests cover areas of optimization, production planning and inventory control, supply chain management, and quality control.