Efficient Least-Loss Algorithm for a Bi-Objective Trim-Loss Problem

Hesham K. Alfares
Systems Engineering Department
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia
alfares@kfupm.edu.sa

Omar G. Alsawafy
Systems Engineering Department
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia
alsawafy@kfupm.edu.sa

***Abstract*— This paper presents a new model and an efficient solution algorithm for a bi-objective one-dimensional trim-loss problem. In the trim-loss – or cutting-stock – 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 several benchmark problems from the literature, the algorithm demonstrated considerable effectiveness in terms of both objectives, in addition to high computational efficiency.**

Keywords*— *Trim-loss problem; one-dimensional cutting-stock problem; multiple-objective optimization models; heuristic algorithms

I. 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, 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. This paper is concerned with a bi-objective one-dimensional cutting stock problem (1D-CSP).

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 (IP) model of the problem is formulated to determine the optimum number of used large objects, and the cutting pattern for each large object. 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 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 II. The mathematical optimization model is formulated in Section III. The two-stage least-loss algorithm (LLA) is described in Section IV. Results of the computational comparisons and benchmarking tests are presented in Section V. Finally, conclusions and suggestions are provided in Section VI.

II. 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. Many previous surveys of literature on various aspects of cutting and packing problems cover relevant trim-loss literature from 1971 to 2004. In this section, the primary focus is on 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 multi-objective 1D-CSP techniques.

Many techniques in the literature deal with bi-objective 1D-CSP. Cerqueira and Yanasse (2009) heuristically minimize setup and trim-loss costs for 1D-CSP by decreasing the number of cut patterns in order to reduce the number of setups. Small items are divided into high demand and low demand groups, and patterns with low trim loss are generated separately for each of the two groups. Kasimbeyli et al. (2011) formulate a pattern-free bi-objective ILP model 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 efficiently solve this ILP model. The same two objectives are used by Golfeto et al. (2009) in a two-population genetic algorithm for 1D-CSP, using one objective for each population. Symbiotic relations between the two sets of objectives and populations are used to produce good solutions.

Berberler and Nuriyev (2010) develop a dynamic programming (DP)-based heuristic approach for solving the 1D-CSP by reducing it to a sequence of smaller sub-problems. A mathematical DP model is formulated and a heuristic solution algorithm is proposed to minimize both the trim loss and the number of cutting patterns. Mobasher and Ekici (2013) combine the two objectives of minimum trim loss (material cost) and minimum cutting patterns (setup cost) into 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.

Multiple-objective 1D-CSP models aim to achieve several simultaneous, often conflicting, objectives. Belov and Scheithauer (2007) use a sequential heuristic to primarily minimize trim loss in a multiple-objective 1D-CSP. Pareto analysis is used to compare solutions in terms of two secondary objectives: minimum cutting patterns (setup cost) and minimum number of open-stacks. Cui (2012) uses the sequential heuristic procedure to develop an algorithm with the main objective of minimizing the total cost. The algorithm identifies the set of non-dominated cutting plans in relation to two auxiliary objectives: minimum cutting patterns, and minimum short stock.

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. Aktin and Özdemir (2009) develop an LP-based two-stage heuristic methodology and apply it in a 1D-CSP case study. In the first stage, an ILP model is used to determine the cutting patterns for minimizing the total trim loss. Another ILP model is then used to determine the cutting plan for minimizing the total manufacturing cost that includes material, labor, setup, and lateness costs.

Although trim-loss problems are NP-complete and generally difficult to optimally solve, optimal solutions have been developed for special cases of the 1D-CSP problem taking advantage of the unique problem structure. Alves and de Carvalho (2008) present one of the few recent optimum solution techniques for 1D-CSP with multiple stock lengths. Valid inequalities are utilized within a branch-and-price-and-cut algorithm to accelerate branching convergence to the optimal solution. Reinertsen and Vossen (2010) present another optimization model and solution technique for 1D-CSP with due date constraints. The model is applied in an industrial case study, incorporating realistic considerations such as order aggregation, multiple stock lengths, and rolling horizons.

Meta-heuristic techniques have been used by several authors for 1D-CSP. Eshghi and Javanshir (2008) apply the ant colony optimization (ACO) heuristic to minimize trim-loss cost in 1D-CSP. In the ACO algorithm, which simulates ant search for food, artificial ants select cutting patterns and determine cutting plans. Golfeto et al. (2009) use genetic algorithms, and Matsumoto et al. (2011) use tabu search. Jahromi and Tavakkoli-Moghaddam (2012) apply two meta-heuristic algorithms, simulated annealing (SA) and tabu search (TS), to the 1D-CSP. Comparing the results on several test problems, they conclude that SA outperforms TS in terms of the objective functions of the solutions.

The above literature review indicates that only Liang et al. (2002) address the same two objectives of this paper: 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 compared with the Liang et al. (2002) EP heuristic.

III. PROBLEM DEFINITION AND MODEL FORMULATION

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.

A. Integer programming model notation

s_i = size (constant length) of all small items in order i, $i = 1, \dots, I$

q_i = quantity of small items in order i, $i = 1, \dots, I$, $q_i \geq 0$ and integer

a_{ij} = number of times size s_i is cut from cutting pattern j, $i = 1, \dots, I$, $j = 1, \dots, J$, $a_{ij} \geq 0$ and integer

w_j = trim loss of cutting pattern j, $0 \leq w_j \leq L$, $j = 1, \dots, J$.

b_j = 1 if cutting pattern j has trim loss ($w_j > 0$), $j = 1, \dots, J$
= 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, $j = 1, \dots, J$

D = allowed trim loss for the given iteration

B. 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:

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

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

C. Constraints

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

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

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:

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

$$L b_j \geq w_j \text{ , } j = 1, \dots, J \tag{6}$$

The above ILP model looks fairly simple, it is quite difficult to optimally solve due to its bi-objective and pure integer nature, in addition to the nonlinearity of the second objective function. Based on the difficulty in obtaining the optimum solution, logical heuristic rules are used to develop a two-stage least-loss solution algorithm, which is presented below.

IV. Least-Loss Algorithm (LLA)

The least-loss algorithm proceeds in two stages as described below.

A. 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 is several widely-used CSP solution methods, including the well-known first-fit-decreasing (FFD) heuristic developed 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. The FFD algorithm is known to be 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 ($s_1 > s_2 > s_3 > \dots > s_n$). Starting with size 1 (largest item), the algorithm first finds all possible combinations of the current size (i) and smaller sizes ($s_i, \dots, s_n$) 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.

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.

$$\text{If } \frac{100(N - N_{\min})}{N_{\min}} > 5\%, \text{ go to Phase 2}$$

Where

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

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

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

B. 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 initially judged not satisfactory. The second stage can be considered as 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, however, the algorithm selects the better of the two solutions produced in stages 1 and 2. Stage 2 of the LLA consists of the two following steps.

Step 1. 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 on the basis of extensive trial-and-error and numerical experimentation.

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

Step 2: Least-loss assignments. Same steps as in Stage 1 are applied on the modified size order defined above.

V. Benchmarking and Computational Analysis

In order to test the least-loss algorithm (LLA), a search was performed for benchmark CSP problems described in the literature. Although there are many 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, 10 problems that have a single large-object length have been used in this paper. 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). Characteristics and size dimensions of the 10 benchmark problems are shown in Liang et al. (2002).

$\alpha + \beta = \chi$ (1) (1)

Computational experiments with the 10 benchmark problems were performed to evaluate the proposed two-stage least-loss algorithm (LLA). The comparison is done in terms of effectiveness, i.e. solution quality, or values of the two objective functions (total trim loss *TL*, and the number of partially-cut large objects *PC*). The proposed LLA 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 the theoretical bound $N_{\min}$ was used to represent the optimal number of large objects, N.

Table 1 shows the values of N obtained by three methods: the optimum bound ($N_{\min}$), Liang et al. (2002), 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, Liang et al. (2002) solution 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 1. Number of large objects used 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 2 compares the solution quality of the least-loss algorithm (LLA) to that of Liang et al. (2002) EP heuristic 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 Liang et al. (2002) solutions. 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 over the Liang et al. (2002) solution as the problem size increases.

TABLE 2. TRIM-LOSS & PARTIALLY-CUT OBJECTS FOR TEST PROBLEMS

Problem no.	***TL***			***PC***		
	Liang	**LLA**	**% less**	**Liang**	**LLA**	**% less**
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

VI. CONCLUSIONS

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 is not practical, 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. Possible extensions of this work include allowing different long-object lengths and stochastic demands, i.e. unknown sizes and quantities of the small items.

ACKNOWLEDGMENT

The authors express gratitude to King Fahd University of Petroleum & Minerals for support and research facilities.

REFERENCES

[1] Aktin T, Özdemir RG (2009). An integrated approach to the one-dimensional cutting stock problem in coronary stent manufacturing. Eur J Oper Res. 196: 737–743.

[2] Alves C, de Carvalho JMV (2008). A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. Comput Oper Res. 35: 1315–1328.

[3] Belov G, Scheithauer G (2007). Setup and open-stacks minimization in one-dimensional stock cutting. INFORMS J Comput. 19: 27–35.

[4] Berberler ME, Nuriyev UG (2010). A new heuristic algorithm for the one-dimensional cutting stock problem. Appl Comput Math. 9: 19–30.

[5] Cerqueira GRL, Yanasse HH (2009). A pattern reduction procedure in a one-dimensional cutting stock problem by grouping items according to their demands. J Comput Interdiscipl Sci. 1: 159–164.

[6] Cui Y (2012). A CAM system for one-dimensional stock cutting. Adv Eng Softw. 47: 7–16.

[7] Eilon S, Christofides N (1971). The loading problem. Manage Sci. 17: 259–268.

[8] Eshgi K, Javanshir H (2008). A revised version of ant colony algorithm for one-dimensional cutting stock problem. Int J Ind Eng-Theory. 15: 341–348.

[9] Garey MR, Johnson DS (1979). Computers and intractability, A Series of Books in the Mathematical Sciences, Vol. 174, Freeman, New York.

[10] Golfeto RR, Moretti AC, de Salles Neto LL (2009). A genetic symbiotic algorithm applied to the one-dimensional cutting stock problem. Pesquisa Operacional. 29: 365–382.

[11] Hinterding R, Khan L, Genetic algorithms for cutting stock problems: with and without contiguity. Springer, Berlin Heidelberg, 1995.

[12] Jahromi MHMA, Tavakkoli-Moghaddam R, Makui A, Shamsi A (2012). Solving an one-dimensional cutting stock problem by simulated annealing and tabu search. J Ind Eng Int. 8: 1–8.

[13] Kasimbeyli N, Sarac T, Kasimbeyli R (2011). A two-objective mathematical model without cutting patterns for one-dimensional assortment problems. J Comput Appl Math. 235: 4663–4674.

[14] Liang KH, Yao X, Newton C, Hoffman D (2002). A new evolutionary approach to cutting stock problems with and without contiguity. Comput Oper Res. 29: 1641–1659.

[15] Mobasher A, Ekici A (2013). Solution approaches for the cutting stock problem with setup cost. Comput Oper Res. 40: 225–235.

[16] Matsumoto K, Umetani S, Nagamochi H (2011). On the one-dimensional stock cutting problem in the paper tube industry. J Sched. 14: 281–290.

[17] Reinertsen H, Vossen TWM (2010). The one-dimensional cutting stock problem with due dates. Eur J Oper Res. 201: 701–711.