

ORIGINAL RESEARCH

Open Access

# Solving an one-dimensional cutting stock problem by simulated annealing and tabu search

Meghdad HMA Jahromi<sup>1\*</sup>, Reza Tavakkoli-Moghaddam<sup>2</sup>, Ahmad Makui<sup>3</sup> and Abbas Shamsi<sup>1</sup>

## Abstract

A cutting stock problem is one of the main and classical problems in operations research that is modeled as LP problem. Because of its NP-hard nature, finding an optimal solution in reasonable time is extremely difficult and at least non-economical. In this paper, two meta-heuristic algorithms, namely simulated annealing (SA) and tabu search (TS), are proposed and developed for this type of the complex and large-sized problem. To evaluate the efficiency of these proposed approaches, several problems are solved using SA and TS, and then the related results are compared. The results show that the proposed SA gives good results in terms of objective function values rather than TS.

**Keywords:** One-dimensional cutting stock problem, Mathematical model, Simulated annealing, Tabu search

## Background

A one-dimensional cutting stock problem (1D-CSP) is one of the famous combinatorial optimization problems, which has many applications in industries. In this problem, the amount of residual pieces of processed stock rolls, called trim loss (i.e., wastage) is a significant objective in most studies that should be minimized. A standard one-dimensional cutting stock problem (S1D-CSP) as a kind of the above problem is known as an NP-complete one (Gradisar et al. 2002). Finding solutions for this problem especially for large-sized problems is extremely difficult. In fact, due to the NP-hardness of this problem, it is computationally intractable to obtain its optimum solution in a reasonable time for medium to large-sized problems, and hence researchers have tried to develop meta-heuristic algorithms for solving such a problem. In fact, applying these algorithms is not easy to reliably state that which of these algorithms is best for the given problem. However, it is possible to identify methods that consistently produce better results compared to others for a certain problem. The assumptions of the S1D-CSP are as follows.

- In this problem, all used stock lengths should be cut to the end in as much as it is possible.

- All stock lengths are identical and there is no difference between the lengths of them.

In a standard cutting problem, each cutting stock is consumed totally, because the remaining is as cutting waste. While in a general one-dimensional stock cutting problem (G1D-CSP), as another kind of the 1D-CSP, stock cutting residuals can be used later. Also, in a G1D-CSP, the storage stock has different lengths. Reduction of cutting wastes is one of the main goals in the cutting process and also one of the basic purposes in the 1D-CSP. A standard one dimensional cutting stock problem (S1D-CSP) and different procedures have been reviewed by many researchers. (Varela et al. 2009) worked on a practical cutting stock problem from a production plant of plastic rolls as a variant of the well-known one dimensional cutting stock. They proposed a heuristic solution based on a GRASP algorithm in their work.

(Cui 2005) proposed an algorithm to produce T-shape cutting patterns applied in the manufacturing of circular and sectional blanks for stators and rotors. The proposed algorithm uses the knapsack algorithm and an enumeration method to determine the optimal combination of blank rows in the strips, the strip numbers and directions in the pattern. A heuristic algorithm for the one-dimensional cutting stock problem with usable left-over is presented by (Cui & Yang 2010).

\* Correspondence: jahromi@iaukhomein.ac.ir

<sup>1</sup>Department of Industrial Engineering, Khomein Branch, Islamic Azad University, Khomein, Iran

Full list of author information is available at the end of the article

Their algorithm consists of two procedures, namely linear programming and sequential heuristic procedures that fulfill the major and remaining portions of the item demands, respectively. This algorithm is able to balance the cost of the consumed bars, the profit from left-overs and the profit from a shorter stocks reduction. (Armbruster 2002) presented a solution approach for the capacitated one-dimensional cutting stock problem under technological constraints by concentrating on solving the pattern sequencing problem occurring from the technological constraints. (Dikili et al. 2007) developed a novel approach for one-dimensional cutting stock problems in ship production via the cutting patterns obtained by the analytical methods at the mathematical modeling stage. The proposed method is able to capture the ideal solution of analytical methods. The main advantage of their proposed method is to guarantee in obtaining an integer solution. Also, the method behaves so that the trim loss is minimized and the stock usage is maximized.

(Liang et al. 2002a) proposed an evolutionary algorithm based on evolutionary programming for cutting stock problems considering contiguity. Two new mutation operators are proposed in their work. Results found by the evolutionary programming are significantly better than or comparable to those found by the genetic algorithm (GA). It is worth to note that evolutionary programming uses mutation as the primary search operator and does not employ any crossover.

(Poldi & Arenales 2009) proposed heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. They reviewed some existing heuristics and proposed others for solving the integer one-dimensional cutting stock problem with multiple stock lengths. Their study deals with the classical one-dimensional integer cutting stock problem. The given objective function is to minimize the waste (i.e., trim loss).

(Umetani et al. 2003) considered the 1D-CSP, in which the number of different cutting patterns is constrained within a given bound. They proposed a meta-heuristic algorithm incorporated with an adaptive pattern generation technique. The proposed meta-heuristic searches a solution with small deviations from the given demands while using the fixed number of different cutting patterns. (Reinertsen & Vossen 2010) considered the problem of cutting stock problems with due date while addressing common cutting considerations, such as aggregation of orders, multiple stock lengths and cutting different types of material on the same machine. In their work, meeting due dates is more important than reducing the scrap. Also, in condition that there is no feasible production plan, their models find a cutting plan

with the minimum weighted tardiness. (Liang et al. 2002b) applied the newest perfect algorithm, which is similar to genetic algorithm, with a mobility function for a cutting stock problem. (Weng et al. 2003) applied the GA to the manufacturing rule oriented cutting of a sectional steel problem for the shipbuilding industry.

(Sung et al. 2004) applied tabu search (TS) to develop an optimization procedure based on the cutting processes in the shipbuilding industry. (Yang et al. 2006) developed the efficient TS method with a mixed objective function for the one dimensional cutting stock problem under a small amount of various cutting conditions. Their proposed TS method consists of a mixed objective function that helps the algorithm to discover the best solution during the one-by-one neighborhood searching processes. (Gradisar et al. 2002) presented an experimental study of various methods aimed at minimizing trim-loss, in their work they mentioned that the LP-based method is possible only when the stock is of the same length or few groups of standard lengths. When all stock lengths are different frequencies, an approach can be used that treat by each item individually.

In addition, (Chen et al. 1996) proposed a simulated annealing (SA) method for the one-dimensional cutting stock problem. A meta-heuristic method based on ACO was presented to solve a one-dimensional cutting stock problem by (Eshghi & Javanshir 2005). In their work, based on probabilistic laws that designed, ants do select various cuts and then select the best patterns. Also, (Dyckhoff 1990) developed a classification scheme for cutting stock problems through a large variety of applications reported in the literature so that classified problems using four characteristics; namely, dimensionality, kind of assignment, assortment of large objects and assortment of small items. He classified the cutting stock problems solutions into two groups: 1) item-oriented approach that treats each item to be cut separately and 2) pattern-oriented approach. So that in this approach at the beginning order lengths are combined into cutting patterns, for which the frequencies that are necessary to satisfy the demands, are determined in a succeeding step. In the second approach, the classic LP-based namely "delayed pattern generation" proposed (Gilmore & Gomory 1961; Gilmore & Gomory 1963) or any other LP-based method is mostly used. (Javanshir & Shadalooee 3) developed a mathematical model for 1D-CSP by defining the virtual cost for the trim loss of each stock. They solved their developed model with SA algorithm. Also, 1D-CSP considered in their work is taken into account as item-oriented.

It is not easy to define exactly which meta-heuristic algorithm is best for which type of problem. However, it

is possible, for a given problem, to identify algorithms that consistently produce better results compared with those results produced by other algorithms. For this reason, in this paper, we compare the efficiency and effectiveness of two well-known meta-heuristic, namely simulated annealing (SA) and tabu search (TS), in producing the cutting plan with the lowest trim loss. In addition, we present a mixed-integer linear programming (MILP) mathematical model for a standard one-dimensional cutting stock problem that minimizes the trim losses of the cutting orders. Because of the NP-hard nature of this problem, the presented mathematical model can find an optimal solution by a standard OR software (e.g., Lingo) for small-sized problems.

This paper is organized as follows. Section 2 presents the mathematical programming model for the S1D-CSP. Simulated annealing (SA) and tabu search (TS) algorithms for the considered problem are proposed in Sections 3 and 4, respectively. The computational experience is represented in Section 5. Finally, conclusion is made in Section 6.

### Mathematical model of S1D-CSP

This section presents a mixed-integer linear programming (MILP) model for a standard one-dimensional cutting stock problem (S1D-CSP). Minimizing the sum of cutting wastes or cutting remaining unusable in satisfying of needs (demands) is the objective function of the presented model.

#### Notations

To formulate the S1D-CSP, the following notations are used.

$i$	order number ( $i = 1, \dots, n$ ).
$j$	stock number ( $j = 1, \dots, m$ ).
$d_j$	stock length.
$tl_j$	stock wastes.
$s_i$	order lengths.
$n_i$	required number of orders with $s_i$ length.
$T$	sum of the cutting wastes in the cutting program.

#### Decision variables

The following variables are used in the presented MILP model.

$x_{ij}$	integer variable, number of orders with $s_i$ length that are cut from stock $j$ .
$y_j$	zero-one variable that equals to one if the stock $j$ is applied in the cutting plan otherwise, equals to zero.

### Mathematical model

The MILP model for a standard one-dimensional cutting stock problem can be defined as follows.

$$\min T = \sum_{j=1}^m tl_j \quad (1)$$

s.t.

$$\sum_{j=1}^m x_{ij} = n_i \quad \forall i \quad (2)$$

$$\sum_{i=1}^n x_{ij} \cdot s_i + tl_j = d_j \cdot y_j \quad \forall j \quad (3)$$

$$y_j \in (0, 1)$$

$$x_{ij} \in \text{integer}$$

Minimizing the total trim loss is the objective function (1) of the model. Constraint (2) guarantees cutting needs of orders with regards to the storage stock. Constraint (3) calculates the cutting waste of each stock in the cutting process.

Due to NP-hard nature of this problem, the computational time increases exponentially and finding optimal solution for large-size problem is extremely difficult. This model is solved by the Lingo software using a branch-and-bound method, and the computational times and the state of the method for 20 randomly generated problems with different sizes are shown in Table 1. Results show finding optimal solution for large-scale problems (15, 16, 17 and 18), in a reasonable time and gradually with increasing the size of the problems (19 and 20), even finding feasible area of solutions, is so hard.

Since finding the optimal solution in a reasonable time for medium to large-sized problems is intractable, a number of researchers have considered developing meta-heuristics to compute near-optimal solutions. It is noticeable that identifying methods consistently producing better results, as compared to others, for a certain problem is very important and interesting. Among the meta-heuristics, simulated annealing (SA) and tabu search (TS) are the ones whose applications have been widely explored for different combinatorial optimization problems. In this work, effectiveness and efficiency of these meta-heuristics for the given problem are compared.

### Simulated annealing for S1D-CSP

(Kirkpatrick et al. 1983) introduced simulated annealing (SA) as a meta-heuristic algorithm in 1983. It draws its inspiration from physical annealing of solids. In this

**Table 1 NP-hard nature of the S1D-CSP**

Problems	Requested orders	State	CPU time (Sec.)
1	9	Global optimum found	1
2	15		1
3	32		2
4	45		4
5	59		7
6	72		10
7	85		28
8	98		59
9	110		607
10	190		2570
11	305	6077	
12	950	15185	
13	1503	58106	
14	1704	79202	
15	1850	Cannot find any global optimum after a 24-hours run	
16	1990		
17	2500		
18	3645		
19	4985	Cannot find any feasible area of solutions after a 24-hours run	
20	5750		

algorithm, it may become stuck in a local optimum. To prevent this from happening and trapping in a local optimum, the algorithm occasionally makes a random change to the solution even through the change does not immediately improve the result. This may allow the program to break free of a local optimum and find a better solution later. If the change does not lead to a better solution, the algorithm will probably undo the change in a short time anyway. SA consists of internal and external loops. Task of the internal loop controls the attaining balance in each temperature and updating temperature.

The Metropolis decision-making loop (i.e., possibility of acceptance of non-improvement) is inserted in the internal loop. According to the decision-making loop, if the current solution  $S_{current}$  cannot improve the objective function comparing with the new neighboring solution  $S_{new}$ ; then, if  $y < e^{-\frac{\Delta}{T}}$  (i.e.,  $\Delta = E(S_{new}) - E(S_{current}) > 0$  and  $y \in [0,1]$ ), the new solution will be accepted. In fact, the acceptance possibility of a neighboring solution is dependent on changing the amount of the objective function, and current temperature. The value of  $T$  starts at initial temperature  $T_0$  and is slowly reduce according to the cooling ratio  $a < 1$  (i.e.,  $T_{k+1} = a.T_k$ ).

Also, the external loop controls the stopping condition of SA. A given SA may iterate for a number of times at

each temperature, called the Markov chain length. The neighborhood search structure generates a new solution from the current candidate solution by slightly changing it. How move from one solution to another one in the solution space is dependent on the structure of designing operators. Attaining to the undiscovered area of exploration space pertain to design efficient operators. The generic flow of SA is given below.

Select an initial temperature  $T_0 > 0$ ;  
 Generate an initial solution,  $S_0$ , and make it the current solution,  $S_{current}$ , and the current best solution,  $S_{best}$ ;  
*repeat*

set repetition counter  $n = 1$ ;

*repeat*

Generates the solution  $S_{new}$  in the neighborhood of  $S_{current}$ ;  
 Calculate  $\Delta = f(S_{current}) - f(S_{new})$ ;  
 If  $(\Delta \leq 0)$  then  $S_{current} = S_{new}$ ;  
 else  $S_{current} = S_{new}$  with the probability  $P(\Delta, T)$ ;  
 if  $(S_{new} < S_{best})$  then  $S_{best} = S_{new}$  ;  
 $n = n + 1$ ;

*until*  $n >$  Markov chain length;

Reduce the temperature  $T$ ;

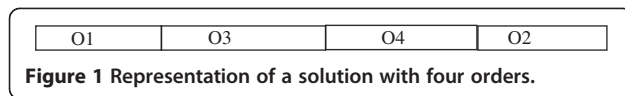
*until* the stopping criterion is met.

### Solution representation

The first step in developing of a meta-heuristic procedure is to design the solution representation. In this paper, with regards to the problem nature, the initial solution is created randomly. This solution is constructed as a  $(1 \times m)$ -array, so that  $m$  equal to the number of orders. Each order placed in a cell randomly as illustrated in Figure 1.

### Neighborhood generation (perturbation)

To search solution space, attaining to undiscovered areas and finding the best solutions, using suitable neighborhood search operators is required. As stated before, designing this operator is dependent on the solution representation. Mechanism of the neighborhood generation should be able to construct the entire solution space. In this paper, in order to generate the neighboring solution, the algorithm randomly changes the place of two orders in the current solution in its array, as shown in Figure 2. This strategy is very simple and guarantees searching the whole of the solution space.



**Figure 1** Representation of a solution with four orders.

**Fitness function**

The objective function considered in this paper minimizes the total trim loss. It is assumed that the stock length is standard with the equal length (e.g., 12 meters considered in this paper). By considering the position of each order in the array from the first (i.e., leftmost) to the end (i.e., rightmost), the total waste is computed. So that all the orders, which the summation of their length is lesser than (or equal to) the length of stock, are assigned to it for cutting and the difference between them considered as the trill loss of that stock and so on, till all orders are assigned.

**Temperature decreasing function**

Temperature is used to compute the acceptance probability of a solution that is worse than the previous one. In the proposed SA algorithm, a geometric function is used as follows.

$$T_k = \alpha T_{k-1} \tag{4}$$

where,  $\alpha$  is a constant less than 1, but it is very close to 1 (i.e., a typical value for  $\alpha$  is chosen between 0.85 and 0.95).  $k$  is the temperature transformations counter. The high amount of  $\alpha$  is as the slow temperature reduction that leads to the best search of neighbors; but, the solution time increases. Adversely, less amount of  $\alpha$  is an indicator of the quick temperature reduction and leads to quicker search in neighbors and sooner coverage in the algorithm.

**Markov chain length**

One of the most important parameters in determining the quality of resultant solutions of SA is the number of searched points in solution space at each temperature, called Markov chain length. In our proposed SA, this counter is denoted by *MarChain\_Length* and used in the

inner loop of SA as a counter. The Markov chain length considered in our proposed SA depends on the problem size (i.e., number of orders).

**Outer counter variable**

The outer counter variable is increased by one, whenever a solution is not selected by probability and shows the number of rejection of perturbed solutions. In fact, if this variable reaches to a pre-determined fixed value as input parameter, the search procedure is stopped or started again depending on the other criteria. Because there is no superior solution exists in the neighborhood, and the search is reached to a near optimal/optimal solution.

**Stopping criterion**

In our proposed SA, the stopping criterion is a number of consecutive times in the cooling process that does not lead to improvement of the objective function as input data.

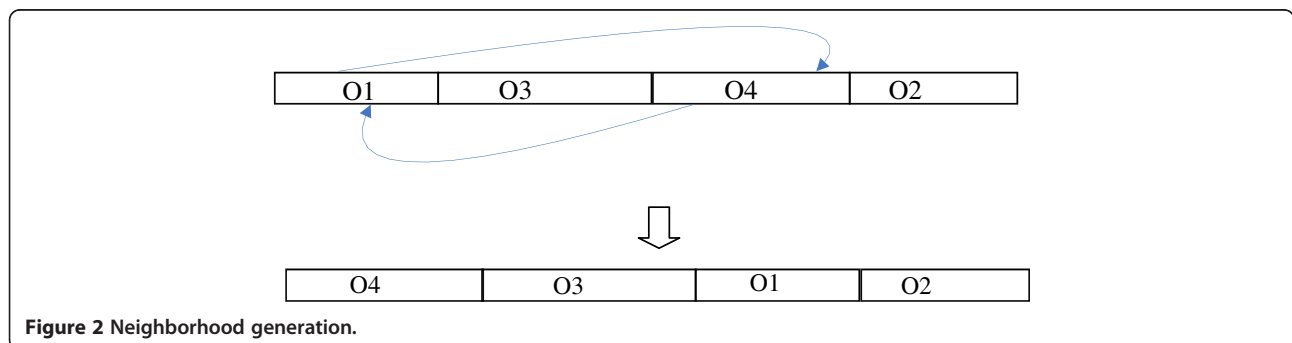
**Tabu search for S1D-CSP**

(Glover 1990) first proposed tabu search (TS) as a higher-level heuristic algorithm used for solving optimization problems. It avoids entrapment in cycles by forbidding or penalizing moves that take the solution, in the next iteration, to points in the solution space previously visited by using the list of prohibited neighboring solutions known as tabu list. Aspiration defined in the tabu search restricts the search from being trapped into a solution surrounded by tabu neighbors. Thus, it helps in the checking condition for the acceptance of solutions, known as a main feature of TS. It also allows search to override the tabu status of the solution and provides backtracking of the recent solutions as they lead to a new path towards a better solution. The generic structure of the TS algorithm is given below.

Step 1. (initialization)

(A). Generate initial solution  $S_{initial}$

(B).  $S_{current} = S_{initial}$



**Figure 2** Neighborhood generation.

- (C). Record the current best known solution by setting  $S_{best} = S_{current}$ , define  $best\_cost = f(S_{best})$
- (D). Set tabu list (H) to empty.

Step 2. (choice and termination)

Determine  $Candidate-N(S_{current})$  as subset of  $N(H, S_{current})$ .  
 Select  $S_{new}$  from  $Candidate-N(S_{current})$ .  
 ( $S_{new}$  is called a highest evaluation element of  $Candidate-N(S_{current})$ .  
 Terminate by a chosen iteration cut-off rule.

Step 3. (update)

Re-set  $S_{current} = S_{new}$ .  
 If  $f(S_{current}) < best\_cost$  or  $f(S_{current}) < Aspiration$  perform Step 1(B)  
 Update the tabu list (H) and Aspiration conditions.  
 Return to Step 2.

**Tabu list**

A tabu list is a set of recently visited solutions and keeps tracking the previously visited solutions in order to restrict the algorithm to revisit the pre-visited solutions. In fact, at every step, the algorithm checks for verifying whether the newly generated solution is lately visited. The tabu list task helps the algorithm to visit more new neighboring solutions.

**Aspiration**

Aspiration is a tabu search variable whose function is to restrict the search from being trapped at a solution surrounded by tabu neighbors. If a solution has a neighborhood of only tabu solutions, the one with objective function value better than aspiration will be chosen to explore further.

**Computational experience**

As mentioned earlier, it is not easy to reliably determine which meta-heuristic method is the best for which problem; but, it is possible to identify methods consistently producing better results, as compared to others for a certain problem. In this section, effectiveness and efficiency of the SA and TS algorithms for the 1D-CSP are considered.

The proposed SA and TS algorithms are coded in Microsoft C#.Net and executed on a Pentium processor running at 2.5 GHz and 2 GB of RAM. The results of solving the 14 small and medium-size test problems solved by branch-and-bound (B&B), SA and TS algorithms are shown in Table 2. This table shows the objective function value (OFV) and the gap between solutions obtained from SA and TS with optimal solutions of the B&B method. The results show that SA and TS algorithms are suitable and both of them even are

**Table 2 Result obtained from SA and TS and branch-and-bound methods**

Problems	B&B method	SA		TS	
	OFV*	OFV	GAP	OFV	GAP
1	4.5	4.5	0.0	4.5	0.0
2	7.5	7.5	0.0	7.5	0.0
3	16	16	0.0	16	0.0
4	22.5	22.5	0.0	22.5	0.0
5	29.5	29.5	0.0	29.5	0.0
6	36	38	5.6	40	11.1
7	34	37	8.8	39	14.7
8	39.2	43	9.7	44	12.2
9	44	45	2.3	50	13.6
10	76	79	3.9	85	11.8
11	122	125.4	2.8	141	15.6
12	285	302	6.0	336	17.9
13	450.9	511	13.3	540	19.8
14	511.2	599.5	17.3	612	19.7

\* Objective function value (meter).

capable to obtain an optimal solution for small-sized problems. Also, these algorithms can find good results for the medium-sized problems so that the gap between them and optimal solution is lower than 20% (the first 14 randomly generated problems are in the small to medium range of size).

To compare the efficiency of these algorithms with each other, the six large-scale problems (15, 16, 17, 18, 19 and 20) are solved and the total stocks used in each solution, the total trim losses and total trim loss ratio of each algorithm are represented in Table 3. The total trim loss ratio can be defined by:

$$\text{Total trim loss ratio} = \frac{\text{Sum of trim loss of all used stocks}}{\text{Stocks used} \times \text{standard length}} \times 100\% \tag{5}$$

It is obvious that SA outperforms TS to get a better minimum number of stocks used or less total trim loss (i.e., total trim loss ratio) so that the mean of the total trim loss ratio of SA for the problems is 6.49 percent while it is 7.45 percent in TS. Figure 3 shows CPU times of two algorithms for the twenty mentioned problems. It is noticeable that more CPU times of the SA algorithm in comparative with TS is due to the internal and external loops and extremely depends on the Markov chain length.

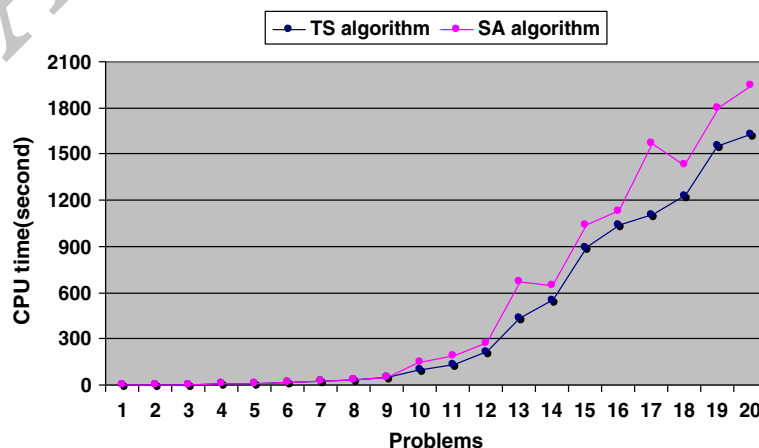
**Table 3 Comparison between SA and TS algorithms**

Problems	Requested orders	SA			TS		
		Stocks used	Trim loss	Total trim loss ratio (%)	Stocks used	Trim loss	Total trim loss ratio (%)
1	9	5	4.5	7.50	5	4.5	7.50
2	15	9	7.5	6.94	9	7.5	6.94
3	32	15	16	8.89	15	16	8.89
4	45	25	22.5	7.50	25	22.5	7.50
5	59	34	29.5	7.23	34	29.5	7.23
6	72	42	38	7.54	44	40	7.58
7	85	51	37	6.05	56	39	5.80
8	98	55	43	6.52	61	44	6.01
9	110	60	45	6.25	69	50	6.04
10	190	111	79	5.93	128	85	5.53
11	305	149	125.4	7.01	173	141	6.79
12	950	501	302	5.02	554	336	5.05
13	1503	850	511	5.01	865	540	5.20
14	1704	951	599.5	5.25	1097	612	4.65
15	1850	1150	751	5.44	1165	1170	8.37
16	1990	1110	897	6.73	1222	1412.4	9.63
17	2500	1350	993	6.13	1480	1955.6	11.01
18	3645	1901	1287	5.64	1984	2192.4	9.21
19	4985	2713	2004.2	6.16	2671	3198	9.98
20	5750	3112	2614	7.00	3221	3930.8	10.17
<b>Average</b>				<b>6.49</b>			<b>7.45</b>

**Conclusions**

In this paper, a mixed-integer linear programming (MILP) model for a standard one-dimensional cutting stock problem (S1D-CSP) has been presented. Due to its NP-hardness, finding feasible solution in a reasonable time, especially for medium to large size problems, is impossible. To solve such a hard problem, two well-

known meta-heuristics based on simulated annealing (SA) and tabu search (TS) have been proposed and developed. To compare the efficiency of these algorithms, a number of test problems have been solved, and the associated results in terms of the objective function (i.e., trim loss) and CPU time have been compared. The results show the efficiency of our proposed SA in finding



**Figure 3** CPU times of the proposed SA and TS algorithms.

cutting plan as its output with lesser trim loss rather than TS. In addition, the results show that TS outperforms SA in terms of the CPU time.

#### Competing interests

The authors declare that they have no competing interests.

#### Authors' contributions

MHMAJ studied the associated literature review and presented the mathematical model. RTM managed the study and participated in the problem solving approaches. AM approved and validate the model and was responsible for revising the manuscript. AS managed and analyzed the required data and conducted the experimental analysis. All authors read and approved the final manuscript.

#### Author details

<sup>1</sup>Department of Industrial Engineering, Khomein Branch, Islamic Azad University, Khomein, Iran. <sup>2</sup>Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran. <sup>3</sup>Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran.

Received: 1 December 2010 Accepted: 3 March 2012

Published: 4 October 2012

#### References

- Armbruster M (2002) A solution procedure for a pattern sequencing problem as part of a one-dimensional cutting stock problem in the steel industry. *Eur J Oper Res* 141(2):328–340
- Chen CLS, Hart SM, Tham WM (1996) A simulated annealing heuristic for the one-dimensional cutting stock problem. *Eur J Oper Res* 93(3):522–535
- Cui Y (2005) A cutting stock problem and its solution in the manufacturing industry of large electric generators. *Comput Oper Res* 32(7):1709–1721
- Cui Y, Yang Y (2010) A heuristic for the one-dimensional cutting stock problem with usable leftover. *Eur J Oper Res* 204(2):245–250
- Dikili AC, Sarlöz E, Pek NA (2007) A successive elimination method for one-dimensional stock cutting problems in ship production. *Ocean Eng* 34(13):1841–1849
- Dyckhoff H (1990) A typology of cutting and packing problems. *Eur J Oper Res* 44(3):145–159
- Eshghi K, Javanshir H (2005) An ACO algorithm for one-dimensional cutting stock problem. *J Ind Eng Int* 1(1):10–19
- Gilmore PC, Gomory RE (1961) A linear programming approach to the cutting stock problem. *Oper Res* 9(3):849–859
- Gilmore PC, Gomory RE (1963) A linear programming approach to the cutting stock problem, Part II. *Oper Res* 11(3):863–888
- Glover F (1990) Tabu search: A tutorial. *Interfaces* 20(4):74–94
- Gradisar M, Resinovic G, Kljajic M (2002) Evaluation of algorithms for one-dimensional cutting. *Comput Oper Res* 29(9):1207–1220
- Javanshir H, Shadalooee M (2007) The trim loss concentration in one-dimensional cutting stock problem (1D-CSP) by defining a virtual cost. *J Ind Eng Int* 3(4):10–19
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220:671–680
- Liang KH, Yao X, Newton C, Hoffman D (2002a) A new evolutionary approach to cutting stock problems with and without contiguity. *Comput Oper Res* 29(12):1641–1659
- Liang KH, Yao X, Newton Y, Hoffman D (2002b) A new evolutionary approach to cutting stock problems with and without contiguity. *Comput Oper Res* 29(12):1641–59
- Poldi KC, Arenales MN (2009) Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Comput Oper Res* 36(6):2074–2081
- Reinertsen H, Vossen TWM (2010) The one-dimensional cutting stock problem with due dates. *Eur J Oper Res* 201(3):701–711
- Sung TC, Weng WC, Yang CT (2004) Optimization of piece arrangement for one-size source cutting problem by tabu search for shipyard. In: Proceeding of the 16th National Conference on Naval Architecture and Marine Engineering. Naval Architecture and Marine Engineering Department, National Cheng-Kung University, Tainan, Taiwan, 5–6
- Umetani S, Yagiura M, Ibaraki T (2003) One-dimensional cutting stock problem to minimize the number of different patterns. *Eur J Oper Res* 146(2):388–402

Varela R, Vela CR, Puente J, Sierra M, González-Rodríguez I (2009) An effective solution for a real cutting stock problem in manufacturing plastic rolls. *Ann Oper Res* 166(1):125–146

Weng WC, Yang CT, Hung CF (2003) The optimization of section steel arrangement for ship construction associated with cutting rule by genetic algorithm. In: Proceeding of the 17th Asian-Pacific Technical Exchange and Advisory Meeting on Marine Structure. National Cheng Kung University, Tainan, Taiwan, 237–246

Yang CT, Sung TC, Weng WC (2006) An improved tabu search approach with mixed objective function for one-dimensional cutting stock problems. *Adv Eng Softw* 37(8):502–513

doi:10.1186/2251-712X-8-24

Cite this article as: Jahromi et al.: Solving an one-dimensional cutting stock problem by simulated annealing and tabu search. *Journal of Industrial Engineering International* 2012 **8**:24.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)