

Aryanezhad, Mir-Bahador; Hashemi, Nima Fakhim; Makui, Ahmad; Javanshir, Hasan

Article

A simple approach to the two-dimensional guillotine cutting stock problem

Journal of Industrial Engineering International

Provided in Cooperation with:

Islamic Azad University (IAU), Tehran

Suggested Citation: Aryanezhad, Mir-Bahador; Hashemi, Nima Fakhim; Makui, Ahmad; Javanshir, Hasan (2012): A simple approach to the two-dimensional guillotine cutting stock problem, Journal of Industrial Engineering International, ISSN 2251-712X, Springer, Heidelberg, Vol. 8, pp. 1-10, <https://doi.org/10.1186/2251-712X-8-21>

This Version is available at:

<https://hdl.handle.net/10419/78591>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by/2.0/>

ORIGINAL RESEARCH

Open Access

A simple approach to the two-dimensional guillotine cutting stock problem

Mir-Bahador Aryanezhad^{1^}, Nima Fakhim Hashemi^{1*}, Ahmad Makui¹ and Hasan Javanshir²

Abstract

Cutting stock problems are within knapsack optimization problems and are considered as a non-deterministic polynomial-time (NP)-hard problem. In this paper, two-dimensional cutting stock problems were presented in which items and stocks were rectangular and cuttings were guillotine. First, a new, practical, rapid, and heuristic method was proposed for such problems. Then, the software implementation and architecture specifications were explained in order to solve guillotine cutting stock problems. This software was implemented by C++ language in a way that, while running the program, the operation report of all the functions was recorded and, at the end, the user had access to all the information related to cutting which included order, dimension and number of cutting pieces, dimension and number of waste pieces, and waste percentage. Finally, the proposed method was evaluated using examples and methods available in the literature. The results showed that the calculation speed of the proposed method was better than that of the other methods and, in some cases, it was much faster. Moreover, it was observed that increasing the size of problems did not cause a considerable increase in calculation time. In another section of the paper, the matter of selecting the appropriate size of sheets was investigated; this subject has been less considered by far. In the solved example, it was observed that incorrect selection from among the available options increased the amount of waste by more than four times. Therefore, it can be concluded that correct selection of stocks for a set of received orders plays a significant role in reducing waste.

Keywords: Cutting stock problem, Trim loss, Two-dimensional cutting, Guillotine cutting

Background

Cutting problems are derived from various industrial processes, for example, textile production systems, glass industries, steel, adhesive tape, wood, paper, etc. (Gonçalves 2007; Ben Messaoud et al. 2008; Leung et al. 2001; Erjavec et al. 2009).

The first research on cutting and packing problems was probably performed by Kantorovich in 1939 and Brooks in 1940, but extensive scientific work in this field began in 1960 as cited in Faina (1999), Dyckhoff (1990), and Javanshir and Shadaloee (2007). Among the most famous studies from that time are those of Gilmore and Gomory (Gilmore and Gomory 1961, 1963, 1965; Alvarez-Valdes et al. 2001; Eshghi and Javanshir 2005). Since the 1960s, many other cutting and packing

problems with the same logic and structure emerged but were given different names in the literature, such as cutting stock and trim-loss problems, bin packing, strip packing, knapsack problems, loading pallets, and loading containers (Faina 1999; Eshghi and Javanshir 2005).

In 1990, Dyckhoff presented a systematic and consistent approach to the integration of different types of problems into a comprehensive typology based on the logical structure of cutting and packing problems. His goal was to assimilate the notations for various uses in the literature and to focus future research on specific types of problems (Dyckhoff 1990).

There are several types of cutting problems based on different characteristics, which Dyckhoff divided into four groups based on their salient characteristics (dimension, type assignment, assortment of large objects, and small items) (Dyckhoff 1990; Dyckhoff and Finke 1992).

Cutting is categorized according to different conditions, such as the cutting style, which can be either

* Correspondence: n_hashemi@ind.iust.ac.ir

[^] Deceased

¹Department of Industrial Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran

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

guillotine or non-guillotine cutting. Guillotine cutting, which is also called edge-to-edge cutting, is a common method for cutting rectangular forms, in which the cutting edge starts on one side and continues to the opposite side (Suliman 2006; Ben Messaoud et al. 2008; Belov and Scheithauer 2006; Mellouli and Dammak 2008).

Another significant condition is whether the items have rotation. Fixed orientation is a state in which there is no possible rotation of the items. Cutting wallpaper or packing articles in a newspaper are examples of this condition (Ben Messaoud et al. 2008; Dyckhoff 1990). This condition should also be considered for the cardboard used for making cartons because of their inter-leaved layers, which play a significant role in the stability of the resulting carton.

In Suliman (2006), a sequential heuristic procedure for a two-dimensional rectangular guillotine cutting stock problem is introduced. The paper provides a three-step solution.

In Gonçalves (2007), the main topic is a two-dimensional orthogonal packing problem, where a fixed group of small rectangles must be fitted into a large rectangle so that most of the material is used, and the unused area of the large rectangle is minimized. The algorithm combines a replacement method with a genetic algorithm based on random numbers; the new algorithm, called 'a hybrid genetic algorithm-heuristic', includes a fitness function and a new heuristic replacement policy.

In Ben Messaoud et al. (2008), the authors focus on guillotine constraints. First, for a cutting problem to be in a guillotine pattern, they provide a necessary condition and then a polynomial algorithm for checking this condition. These methods are useful when the raw materials for cutting are expensive. They also claimed that their study is the first to formulate the guillotine constraints in terms of linear inequalities. In fact, an integer programming formulation is provided that considers the guillotine constraints explicitly. However, an adaptation of their algorithm enables it to convert an initial non-guillotine pattern to a guillotine pattern and to recognize all non-guillotine regions in a given pattern.

In Alvarez-Valdés et al. (2002), a number of heuristic algorithms for two-dimensional cutting problems (on large scales) are developed. In this study, there is a large primary stock that has to be cut into smaller pieces so as to maximize the value of the pieces. They developed a greedy randomized adaptive search procedure (GRASP), which is a fast method that provides good-quality answers. They also developed a Tabu search algorithm, which was more complex but gave better results in reasonable computational time. They also stated that the GRASP method should be used if quick solutions are needed, whereas the Tabu search algorithm should be selected if quality is more important.

In Faina (1999), the two-dimensional cutting stock problem is solved using a general optimization algorithm and simulated annealing. The algorithm includes the guillotine and non-guillotine constraints. The author also noted that for large numbers of items, the NON-GA is far superior to the GA because it obtains better cutting patterns in almost the same time.

Assembly line balancing problems have attracted attention for years. These problems were considered in around 1950s. The studies of Bryton in 1954 and Salveson in 1955 as cited in Malakooti (1991, 1994), Fathi et al. (2011), and Roy and Khan (2010, 2011a, 2011b) can be referred to as examples. In general, assembly line balancing seeks to assign a number of tasks to workstations considering precedence (priority) relations so that a product is ready in a given cycle time and efficiency is maximized (Malakooti 1991, 1994; McMullen and Tarasewich 2003; Capacho and Pastor 2008; Fathi et al. 2011; Roy and Khan 2010, 2011a, 2011b).

Most of the methods developed for two-dimensional cutting stock problems are difficult, time-consuming, and costly. In addition, most are suitable only for special kinds of problems, so most factories are not interested in using such methods but rely on traditional procedures (based on the author's personal experience in the carton industry). The main goal of this paper is to obtain a simple and practical approach by finding and exploiting a meaningful relation between cutting stock problems, assembly line balancing problems, and number of workstation problems.

The other subject that we will discuss, which has been neglected to some extent in the literature thus far, is the problem of selecting the best stock size from the available stocks in the market and the effect of size selection on the level of waste.

The contents below are organized as follows: the 'Problem description' section describes the problem; the 'Suggested cutting approach' section suggests the cutting approach and describes the algorithm; the 'Minimum number of stocks' section determines the minimum number of stocks required; the 'Stock selection' section focuses on the stock selection; the 'Results and discussion' section reports the experimental results; and the 'Conclusions' section gives the conclusions and suggests subjects for future research.

Methods

Problem description

Two-dimensional regular cutting stock problem

Packing problems or two-dimensional cutting problems are given in the following form: given a series of large rectangles of a standard size (stocks) and a series of small rectangles (items or pieces), the aim is to cut or

pack all items while minimizing the waste (Ben Messaoud et al. 2008).

Assumptions

In this research, we make the following assumptions:

- a) Cuts are guillotine cuts.
- b) The stocks and items are rectangular.
- c) The orientation is fixed, and the items are not rotated.
- d) The assignment is of the third kind (all items, selection of objects) (Dyckhoff and Finke 1992; Wäscher et al. 2007) because all items must be cut to serve all demands.
- e) The model is proposed under deterministic conditions (inputs are deterministic).
- f) The remaining sheets in each size, when all demands have been met, are considered as waste, although they may be used for further orders. (Because the stocks are not assumed to be in rolls and are separated, if there are no orders left, the remaining parts of any size are considered temporarily as waste). This assumption causes the waste percentage to be higher, but possibly only temporarily; as new demands are received and new parts are cut from the waste, the waste percentage decreases.

Suggested cutting approach

As observed in Neapolitan and Naimipour (2004), packing and cutting stock problems are considered to be in the field of knapsack problems, which are non-deterministic polynomial-time (NP)-hard problems. One of the methods discussed in this book is approximation algorithm design. This algorithm is used for a NP-hard optimization problem. There is no proof that it offers an optimal solution, but it offers solutions that are logically close to the optimal solution. Neapolitan and Naimipour (2004) present an approximation algorithm for the bin-packing problem, which, in fact, is equivalent to a one-dimensional cutting stock problem; this simple strategy is called 'first-fit', and the limit of the closeness of the approximate solution to the optimal solution is well defined.

The theorem is as follows:

Let 'opt' be the optimum number of bins for a packing problem and 'approx' be the number of bins used by the non-descending first-fit algorithm; then,

$$\text{approx} \leq 1.5 \times \text{opt}. \quad (1)$$

The above theory was proved in the ninth chapter of the book by Neapolitan and Naimipour.

In Dyckhoff (1990), the author states that the first-fit-decreasing algorithm (FFD) consumes up to 22.3% more stock than under-optimized conditions and that the unnecessary waste is no more than 18.2%. He then considers a condition under which the waste is reduced to 1.9% and

finally observes that if the number of items tends to infinity, the average amount of waste tends to zero.

In Dósa (2007), the author first reviews the previous bounds, then observes that minimizing the constant in the equation has been an open question over the years, and then states,

$$\text{FFD}(I) \leq 11/9 \text{OPT}(I) + 6/9. \quad (2)$$

He also notes that this is a tight bound.

As described in Neapolitan and Naimipour (2004), these methods are not subject to specific limitations and complexities, and normally they provide a convenient solution. The main advantage of cutting stock problems is that to evaluate the quality of a solution, there is no need to know the exact optimum solution, as the waste percentage itself describes the quality of the solution.

In this part, we suggest an easy way to decrease trim loss and reduce stock sheet usage. We also examine the guillotine form of two-dimensional cutting stock problems. According to the definition of guillotine cutting, it seems that the waste in this form of problem is greater than or equal to that in the non-guillotine form, as the guillotine cut imposes a limitation and, as we know, as the limitations increase, if the solution space does not shrink, at best, it will be constant and not grow.

The first principle that comes to mind in the guillotine form is that the width of the pieces in each strip should be equal or as close as possible to the width of the strip, so as to reduce the waste at the edges of the strips.

The cutting problems in this research are assumed to be almost the same as the balancing problems. Therefore, we have tried to take advantage of their methods as much as possible and adapt them as necessary. In this part, we use a method similar to the ranked positional weight (RPW) proposed by Helgeson and Bernie in 1961 for balancing problems as cited in Capacho Betancourt (2007).

The RPW method is also very similar conceptually to the FFD method, which has been used in one-dimensional cutting stock problems (Javanshir and Shadalooee 2007). We have therefore tried to suggest a suitable method for two-dimensional guillotine cutting problems using the above concepts. As mentioned before, we have tried to suggest a simple, applicable, quick, and easy calculation method that is user-friendly and easily applicable for industry.

Similarity of the RPW and FFD methods

The RPW method is a method for the assignment of tasks to workstations in line balancing problems. The stages of the RPW method are as follows:

1. First step: Calculation of weights for each task according to their positions.

2. Second step: Descending arrangement and ranking of the tasks according to their positional weights.
3. Third step: Initial assignment of tasks to stations according to the cycle time and precedence from the beginning of the list.

The FFD method is conceptually similar to the RPW method. The approach suggested in this paper for the two-dimensional guillotine cutting problem will make use of these concepts and offer some suggestions. It is necessary to mention that the precedence mentioned in the third step of the PRW method also exists in the

two-dimensional guillotine cutting problem as a different form to be considered. In fact, when creating a strip in two-dimensional guillotine cutting, the width of each piece cut from the strip should be less than or equal to the width of the strip.

Suggested approach for two-dimensional guillotine cutting
 The assumed problem is two-dimensional guillotine cutting, which we can describe in terms of the three steps of the RPW method as follows:

1. First step: Assigning priorities to the pieces to be cut (weights).

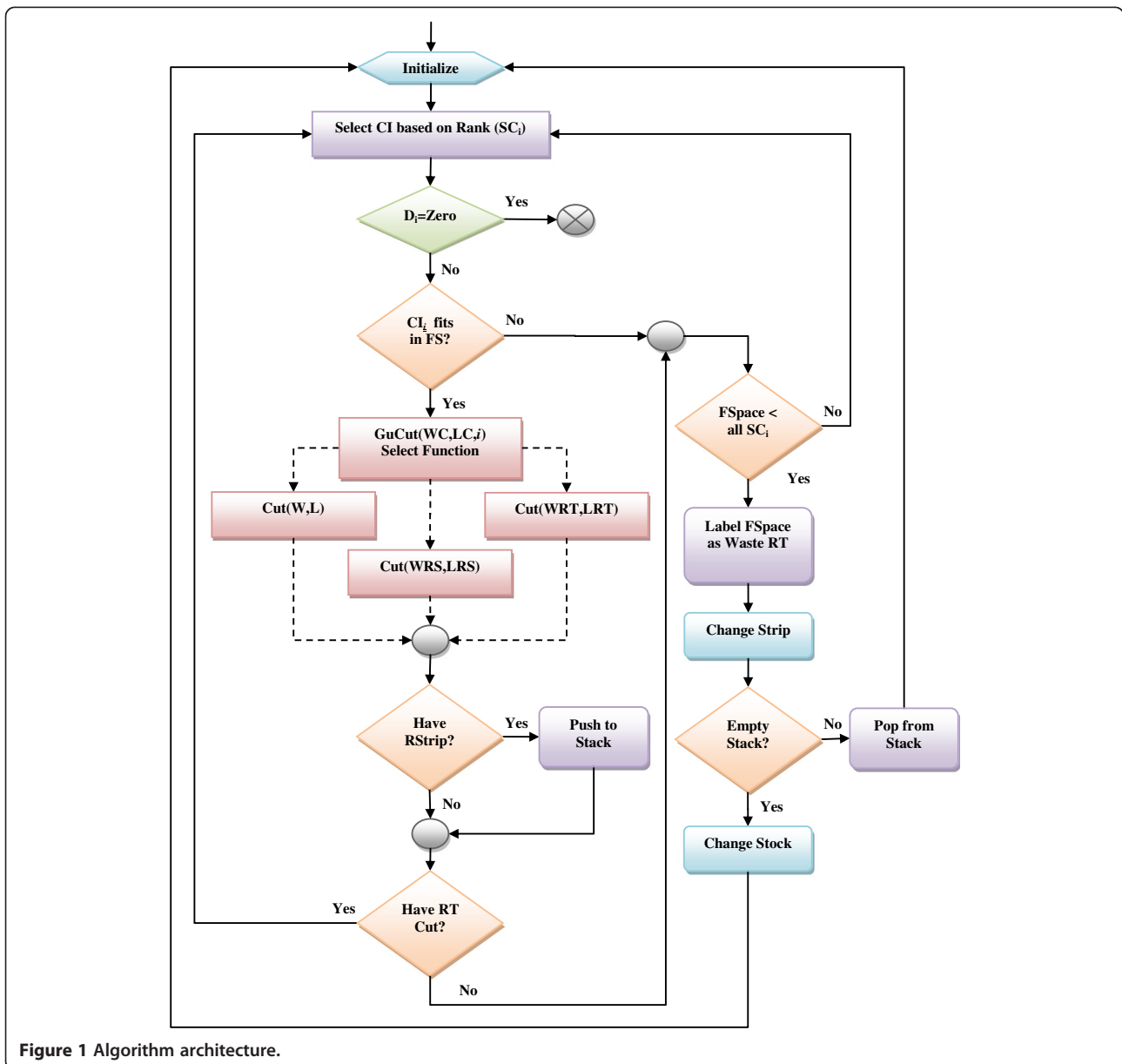


Figure 1 Algorithm architecture.

Table 1 Example specifications

Reference	Example	Stock size	Items (quantity)	Types
Burke, Kandall, Whitwell (chapter 3)(Whitwell 2004)	N1	40 × 40	10	10
	N2	30 × 50	20	20
	N3	30 × 50	30	30
	N4	80 × 80	40	40
	N5	100 × 100	50	50
Burke, Kandall, Whitwell (chapter 4)(Whitwell 2004)	MT1	400 × 600	100	12
	MT2	300 × 900	150	10
	MT3	400 × 900	200	23
	MT4	600 × 1,300	300	24
	MT5	800 × 1,400	500	47

- Second step: Descending arrangement of the pieces by priority (weights).
- Third step: Cutting the pieces from the beginning of the list produced in the previous step, considering the length and width of the stock.

Suggested algorithm implementation

In this algorithm, W is the width of stock; L , length of stock; N , quantity of needed stocks; w_i , width of the pieces ($w_i \leq W \forall i = 1, 2, \dots, n$); l_i , length of the pieces ($l_i \leq L \forall i = 1, 2, \dots, n$); and d_i , number of demands for each kind of pieces.

Stages of the algorithm. The following are the stages of the algorithm:

- Initializing the primary quantities:
 In this step, the dimensions of the stock, the requested pieces, and the demand for each item are input: W, L, w_i, l_i, d_i ($i = 1, 2, \dots, n$)
- Primary calculations:

- Calculating the areas of the pieces.

$$a_i = w_i \times l_i, (i = 1, 2, \dots, n) \quad (3)$$

- Multiply the demand by the area.

$$a'_i = a_i \times d_i, (i = 1, 2, \dots, n) \quad (4)$$

- Calculating the total area of the demand.

$$A_t = \sum_{i=1}^n a'_i \quad (5)$$

- Assigning priorities to the pieces to be cut:
 Here, all of the demands will be arranged in descending order by the width of the pieces. If the widths of two pieces are the same, then the longer piece will be chosen. In this step, the priorities can also be assigned by the sum of the widths and lengths or by the area. While the cuts in this paper are assumed to be guillotine, it makes sense to assign priorities by the widths of the demand pieces because most of the waste in the guillotine cuts is created in the strips and while trimming, and in this suggested approach, the widths of the pieces in each strip are equal or as close as possible. It is possible that assigning priorities based on the sum of widths, lengths, or areas might produce better results in some non-guillotine cases. It is even possible to use a

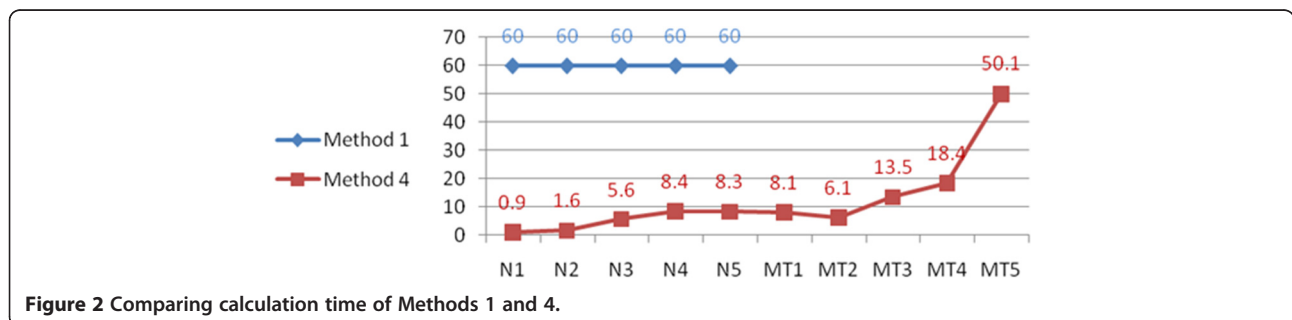


Figure 2 Comparing calculation time of Methods 1 and 4.

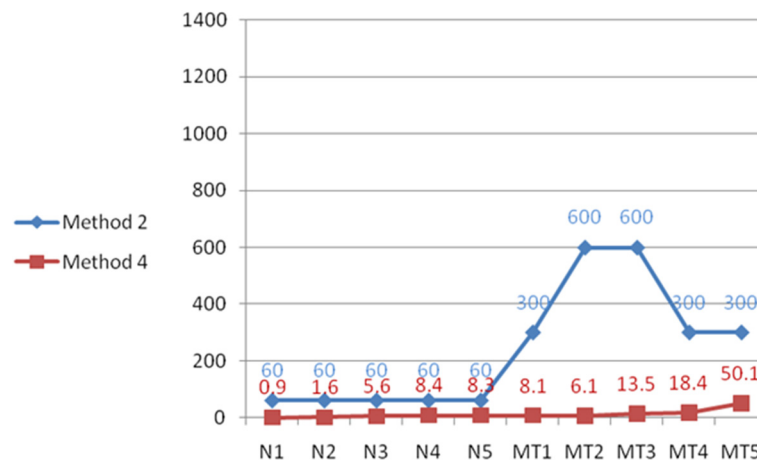


Figure 3 Comparing calculation time of Methods 2 and 4.

combination of one of these methods and the bottom left method.

In two-dimensional guillotine cutting, the widths of the pieces should also be controlled in each cut because the widths of the pieces should be less than or equal to the width of the strip created in the guillotine cuts. For this reason, assigning priority by the width of the pieces also has the additional advantage of reducing the controls in each cut. In this case, there is no need to compare the widths of the pieces with the width of the strip because surely they are less than or equal to the width of the strip, and this increases the speed of the algorithm. Finally, a list of cutting priorities is created.

4. Loading the stock with width W and length L .
5. The first undone item in the cutting priority list (CPL) is loaded, and its width is cut from the

remaining stock if possible. Then, the item's lengths are cut out of the cut piece. If the width or the length of the demanded pieces was greater than the width or the length of the remaining stock, proceed to the next step.

6. Repeat step 5 for the other demands in order of CPL. If the width or the length of the remaining stock is less than all remaining demands, then go to step 4; if not, go to the next step.
7. If all of the demands have been completed, the algorithm stops, performs final calculations, and reports the result.

Algorithm performance criteria. The algorithm performance is defined as the ratio of total area of the demanded pieces to the total area of the stocks consumed in order to fulfil the demand. This ratio is the

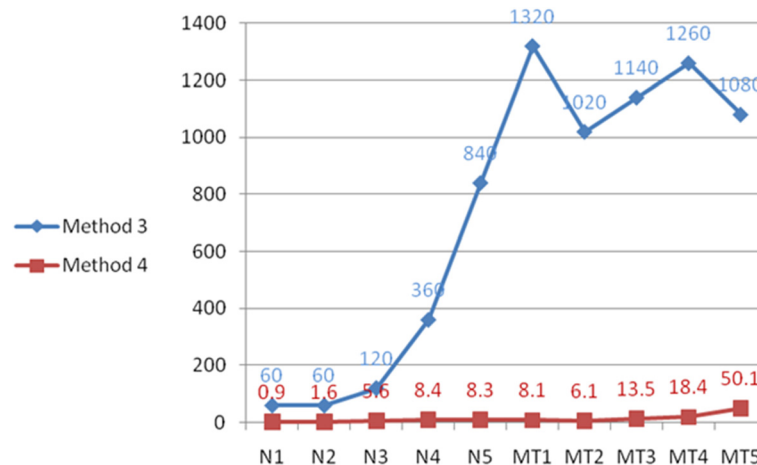


Figure 4 Comparing calculation time of Methods 3 and 4.

same as the efficiency in balancing problems, which is derived as follows:

$$\text{Efficiency} = \frac{\sum_{i=1}^n t_i}{m \times t_c}, \quad (6)$$

where t_i is the processing time of task, t_c is the cycle time, and m is the number of workstations. The efficiency for the proposed approach is calculated as follows using the same method:

$$E = \frac{\sum_{i=1}^n a_i \times d_i}{N \times W \times L} \times 100. \quad (7)$$

We could also define the percentage of waste as follows:

$$\% \text{Wastage} = \frac{(N \times W \times L) - \sum_{i=1}^n a_i \times d_i}{N \times W \times L} \times 100. \quad (8)$$

Proposed algorithm architecture

The first step in the algorithm (Figure 1) is the Initialization phase, in which all the variables are set according to the current iteration and the initial parameters. Then, an item is selected based on the priority table for cutting. This step is represented in the Select Current Item part of the flowchart. Following this step, the Termination Condition will be checked to determine whether all the demands have been satisfied. The algorithm stops when there are no more demands left. In cases where there are still unsatisfied demands, the process continues to compare the dimensions of the Current Item to the Free Space left on the strip. If the cutting is feasible, then the Cut Function will be called.

There are three different scenarios in which the main cutting function can be called. In every run, one of these cases occurs. The first case is when the free space is equal to the raw stock (W, L) size. From the second iteration and after the cutting procedure, there will be two remainder strips from the stock. The width and length of the upper strip are represented by WRS and LRS , respectively. In every cycle, newly generated strips are placed on the top of the stack, which forms the second scenario. Conversely, the lower strips are used to produce new items in the third scenario. They will be used until there are no more feasible cuts, which lead to waste items. Then, the control flow will change to the stack items and the pop function will be called. The cutting algorithm continues to run until the control flow reaches the waste items. In a situation where there are no strips and the stack is empty, a new raw stock will be

used. The above algorithm has been implemented using the C++ language.

The design and implementation of the applied algorithm for solving such problems has a direct influence on the reduction of calculation time. The software architecture presented in this paper can be extended to different cutting approaches.

Minimum number of stocks

In this section, we will determine the minimum number of stocks required. When our organization receives a list of demands, the first step is to provide the stock sheets. This step should be done as soon as possible so that the process can start without delay. Additionally, in this case, the cutting problem is treated as a balancing problem (number of workstations).

The first kind of balancing problem, the simple assembly line balancing problem (SALBP-1), assumes that the cycle time is given and the target is to minimize the number of workstations (Capacho Betancourt 2007).

We now assume that the cutting problem is a SALBP-1, in which the area of the stock corresponds to the given cycle time in the SALBP-1. Balancing problems are problems of calculating a limit for the number of workstations.

The lower bound on the number of workstations is calculated using the following formula:

$$m_{\min} = \left\lceil \sum_{i=1}^n t_i / t_c \right\rceil, \quad (9)$$

where m_{\min} is the lower bound for the minimum number of workstations, t_i is the required time for each task, and t_c is the cycle time.

Using the above formula, we can calculate the lower bound on the number of stocks:

$$S_{\min} = \left\lceil \sum_{i=1}^n a_i \times d_i / W \times L \right\rceil, \quad (10)$$

where S_{\min} is the lower bound on the number of stocks, a_i is the area of each item, d_i is the number of demand for each item, and W and L are the width and length of the stock, respectively.

We can thus calculate the minimum number of required stocks. This lower bound is valid for every cutting method.

The upper bound for the number of workstations in balancing problems is

$$m_{\max} = n, \quad (11)$$

where ' n ' refers to the number of tasks, but this upper bound is not applicable for the required number of

stocks. (In further studies, the applicable upper bound based on this approach could be examined).

Stock selection

In this section, the main target is to select the proper stock. Stock producers and suppliers generally produce stocks in various sizes, and companies have several choices available. Therefore, we have exploited this fact and assumed that depending on demands received, the companies could choose the stock size that ultimately results in the least waste.

Suppose the organization has received an order. To satisfy this order, the stocks satisfying the following condition could be chosen: $\{W \geq w_i, L \geq l_i \forall i = 1, 2, \dots, n\}$.

The waste is calculated for each of the available stocks in the market for which this condition applies, and finally, the stock with the lowest waste is selected.

$$\text{Best stock} = \text{Min} \{ \%wastage_1, \%wastage_2, \dots, \%wastage_k \},$$

where $wastage_k$ is the waste of the k th-type stock.

The important point is that this criterion is used under the assumption that the price is calculated based on the area or weight of the sheets. (In weight mode, it is assumed that the weight is proportional to the area). Otherwise, the conditions will be different. For example, if the price is not calculated based on area or weight, but each type of stock has a unique price (not related to area or weight directly), then a different selection criterion must be defined, which will be described later on.

Here, the selection criterion is

$$\text{Best stock} = \text{Min}\{(N_1 \times C_1), (N_2 \times C_2), \dots, (N_k \times C_k)\},$$

where C_k is the k th-type stock cost and N_k is the number of the k th-type stocks used.

Results and discussion

In this section, computational results for a number of examples are given to evaluate the algorithm's efficiency. The results were derived by running the software on a Core 2 Duo 2.53 GHz PC with 4 GB of RAM.

Approach evaluation

In this section, we evaluate the numerical results obtained by the approach discussed in the 'Suggested cutting approach' section of this article and by some pre-existing methods in the literature on rectangular sheets and parts (Whitwell 2004). Ten examples including N1, N2, ..., N5 and MT1, MT2, ..., MT5 were considered. The specifications of these examples including sheet dimensions and the number and type of pieces which indicated the size of problems are given in Table 1.

All the examples were solved using the proposed approach, and they were compared with the results of

three other methods in Figures 2, 3, and 4 (Method 1: metaheuristic bottom-left-fill (MBLF); Method 2: best-fit (BF) + MBLF; Method 3: interactive approach (IA); Method 4: the proposed method of this paper).

The comparison criterion was the calculation speed. Each of the examples was placed on the horizontal axis, and their solving time (in seconds) was specified on the vertical axis.

As can be observed in Figure 2, the calculation time of the proposed method (Method 4) is much lower than that of Method 1 in five initial examples. For Method 1 in the following five examples, no results have been reported in the literature.

Also, it can be seen in Figures 3 and 4 that the proposed method (Method 4) conducts and ends the calculations faster than Methods 2 and 3. Moreover, the calculation time of the proposed method (Method 4) shows no noticeable increase by increasing the size of problems. It should be mentioned that the time declared for the proposed method (Method 4) includes the time spent for generating the report in addition to the solving time.

Influence of stock selection on amount of waste

We present an example of the influence of stock selection on the amount of waste produced. Suppose demands are received as in Table 2.

Suppose there are 20 different kinds of stock available on the market, naturally, the best choice is the stock size that leads to the least waste. The waste percentages for these 20 stocks are listed in Table 3.

As seen here, the choice of stock for a series of demands has a noticeable influence on the waste; the

Table 2 List of items and demands

<i>i</i>	<i>w_i</i>	<i>l_i</i>	<i>d_i</i>
1	25	40	16
2	60	120	11
3	10	10	18
4	25	50	14
5	35	50	10
6	15	20	25
7	70	95	11
8	15	25	24
9	58	70	15
10	5	10	6
11	42	60	6
12	15	15	16
13	40	60	10
14	30	65	15
15	20	35	4

Table 3 Waste percentage for each kind of stock

N_i	Stock size	S_{min}	S_{used}	%Waste	Best choice
1	70 × 120	43	50	14.8524	
2	80 × 120	38	49	23.9753	
3	100 × 120	30	37	19.455	
4	100 × 150	24	30	20.5289	
5	100 × 180	20	24	17.2176	
6	100 × 200	18	19	5.88947	√
7	100 × 220	17	19	14.445	
8	100 × 230	16	18	13.6184	
9	100 × 250	15	16	10.595	
10	100 × 260	14	15	8.30256	
11	100 × 320	12	12	6.86979	
12	120 × 120	25	28	11.3046	
13	140 × 220	12	14	17.064	
14	140 × 260	10	11	10.6843	
15	140 × 400	7	7	8.77041	
16	200 × 310	6	7	17.5991	
17	200 × 330	6	6	9.69192	
18	200 × 340	6	6	12.348	
19	220 × 320	6	6	15.3362	
20	300 × 400	3	4	25.4958	

example shows that an improper stock selection could increase the waste by more than four times. Therefore, it seems logical that factories should choose the stocks that lead to the least waste. This means that it is also important to consider the raw material selection. It is important to remember that the waste amounts reported in the table could be temporary, as it is possible to decrease the waste by processing further orders.

Conclusions

Considering that cutting stock problems are NP-hard, heuristic methods, like the one presented in this paper, can be efficiently applied for such problems. As was observed before, the proposed method did not have specific complexities of these types of problems. Also, it was observed that it had better speed in solving the problems in comparison with other methods.

The proposed algorithm was designed and implemented using C++ language in a way that it recorded the report of all the functions and, at the end, the user accessed all the information related to the cutting. At the same time, it can be easily used by operators. Most industries and workshops are not interested in using scientific methods since they are time-consuming, costly, complex, and limited. As described above, the proposed approach is a simple and quick method, which makes it easier to use.



Figure 5 Professor Mir-Bahador Aryanezhad.

Furthermore, better selection of sheets from among the available options (options on the stock) has a considerable effect on decreasing the amount of waste. It was observed in the sample solved above that, by correct selection of the stock size, the amount of sheet waste decreased by more than one-fourth.

Presenting simple and combinatorial approaches (hybrid methods) for non-guillotine problems can be considered in future studies. Also, meta-heuristic methods can be applied in this regard.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MBA proposed, designed, and managed the study. NFH worked on the study details (proposed the algorithms and approaches and carried them out) and drafted the manuscript. AM and HJ helped and participated in its design and coordination. All authors read and approved the final manuscript.

Authors' information

MBA was a professor, NFH is an M.Sc. graduate, and AM is an associate professor in the Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran. HJ is an assistant professor in the Department of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran.

Acknowledgements

Here, Nima Fakhim Hashemi would like to thank all the honorable teachers and professors during his study years, especially Professor Aryanezhad who recently passed away. He was a kind, well-educated, and morally distinctive professor who was always guiding and encouraging his students. His memories will always be with us. May God bless his soul (Figure 5).

Author details

¹Department of Industrial Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran. ²Department of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran 11518-63411, Iran.

Received: 7 June 2011 Accepted: 5 March 2012

Published: 18 September 2012

References

- Alvarez-Valdes R, Parajon A, Tamarit JM (2001) A computational study of heuristic algorithms for two-dimensional cutting stock problems. 4th metaheuristics international conference (MIC'2001) 16–20, Porto
- Alvarez-Valdés R, Parajón A, Tamarit JM (2002) A tabu search algorithm for large-scale guillotine (un)constrained two-dimensional cutting problems. *Comput Oper Res* 29(7):925–947
- Belov G, Scheithauer G (2006) A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. *Eur J Oper Res* 171(1):85–106
- Ben Messaoud S, Chu C, Espinouse M (2008) Discrete optimization characterization and modelling of guillotine constraints. *Eur J Oper Res* 191(1):112–126
- Capacho Betancourt L (2007) ASALBP: the alternative subgraphs assembly line balancing problem. Technical University of Catalonia, Thesis (Ph.D.)
- Capacho L, Pastor R (2008) ASALBP: the alternative subgraphs assembly line balancing problem. *Int J Prod Res* 46(13):3503–3516
- Dósa G (2007) The tight bound of first fit decreasing bin-packing algorithm is $FFD(1) \leq 11/9 OPT(1) + 6/9$. *Lect Notes Comput Sci* 4614:1–11
- Dyckhoff H (1990) A typology of cutting and packing problems. *Eur J Oper Res* 44(2):145–159
- Dyckhoff H, Finke U (1992) Cutting and packing in production and distribution: a typology and bibliography. Physical-Verlag, Heidelberg
- Erjavec J, Gradišar M, Trkman P (2009) Renovation of the cutting stock process. *Int J Prod Res* 47(14):3979–3996
- Eshghi K, Javanshir H (2005) An ACO algorithm for one-dimensional cutting stock problem. *J Ind Eng Int* 1(1):10–19
- Faina L (1999) An application of simulated annealing to the cutting stock problem. *Eur J Oper Res* 114(3):542–556
- Fathi M, Jahan A, Ariffin MKA, Ismail N (2011) A new heuristic method based on CPM in SALBP. *J Ind Eng Int* 7(13):1–11
- Gilmore PC, Gomory RE (1961) A linear programming approach to the cutting-stock problem. *Oper Res* 9(6):849–859
- Gilmore PC, Gomory RE (1963) A linear programming approach to the cutting-stock problem—part II. *Oper Res* 11(6):863–888
- Gilmore PC, Gomory RE (1965) Multistage cutting stock problems of two and more dimensions. *Oper Res* 13(1):94–120
- Gonçalves JF (2007) A hybrid genetic algorithm-heuristic for a two-dimensional orthogonal packing problem. *Eur J Oper Res* 183(3):1212–1229
- 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):51–58
- Leung TW, Yung CH, Troutt MD (2001) Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem. *Comput Ind Eng* 40(3):201–214
- Malakooti B (1991) A multiple criteria decision making approach for the assembly line balancing problem. *Int J Prod Res* 29(10):1979–2001
- Malakooti B (1994) Assembly line balancing with buffers by multiple criteria optimization. *Int J Prod Res* 32(9):2159–2178
- McMullen PR, Tarasewich P (2003) Using ant techniques to solve the assembly line balancing problem. *IIE Transaction* 35:605–617
- Mellouli A, Dammak M (2008) An algorithm for the two-dimensional cutting-stock problem based on a pattern generation procedure. *Inf Manag Sci* 19(2):201–218
- Neapolitan R, Naimipour K (2004) Foundations of algorithms using C++ pseudocode, 3rd edn. Jones and Bartlett Publishers, Sudbury
- Roy D, Khan D (2010) Assembly line balancing to minimize balancing loss and system loss. *J Ind Eng Int* 6(11):1–5
- Roy D, Khan D (2011a) A new type of problem to stabilize an assembly setup. *Manag Sci Lett* 1(3):271–278
- Roy D, Khan D (2011b) Optimum assembly line balancing by minimizing balancing loss and a range based measure for system loss. *Manag Sci Lett* 1(1):13–22
- Suliman SMA (2006) A sequential heuristic procedure for the two-dimensional cutting-stock problem. *Int J Prod Econ* 99(1–2):177–185
- Wäscher G, Haußner H, Schumann H (2007) An improved typology of cutting and packing problems. *Eur J Oper Res Research* 183(3):1109–1130
- Whitwell G (2004) Novel heuristic and metaheuristic approaches to cutting and packing. University of Nottingham, Thesis (Ph.D.)

doi:10.1186/2251-712X-8-21

Cite this article as: Aryanezhad et al.: A simple approach to the two-dimensional guillotine cutting stock problem. *Journal of Industrial Engineering International* 2012 **8**:21.

Submit your manuscript to a SpringerOpen® 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