International Journal of Industrial Engineering, 13(2), 198-206, 2006.

AN IMPROVED HEURISTIC FOR THE TWO-DIMENSIONAL CUTTING STOCK PROBLEM WITH MULTIPLE SIZED STOCK SHEETS

Ahmed El-Bouri

Department of Mechanical and Industrial Engineering
Ryerson University,
350 Victoria St.
Toronto, Canada M5B 2K3
aelbouri@ryerson.ca

Jinsong Rao, Neil Popplewell, and S. Balakrishnan

Department of Mechanical and Industrial Engineering
University of Manitoba
15Gilson St.
Winnipeg, Canada R3T 5V6

This paper deals with the problem of cutting multiple sized, rectangular stock sheets into smaller rectangular order pieces to satisfy a given bill of material with minimum trim loss. A new heuristic procedure is devised that finds an effective stock sheet selection sequence, given that the layout procedure used for individual sheets is known. Results for randomly created test problems are compared with those from three previously published procedures. The new heuristic is shown to give a balanced trade-off between trim loss reduction and computational effort, especially as the number of available stock sheet sizes increases.

Significance: The approach presented in this paper significantly reduces trim loss in sheet cutting operations, with only modest increases in computational effort in comparison to the quickest among a selection of existing heuristics.

Keywords: Two-dimensional cutting stock problem, Multiple stock sheets, Bin packing, Trim loss, Heuristic.

(Received 19 May 2004; Accepted in revised form 25 October 2005)

1. INTRODUCTION

In the metal, leather, glass, electronic, shipbuilding, lumber and other industries, raw material is usually stocked in standard sizes which have to be cut into the smaller pieces needed to fulfil a given Bill of Material (BOM). A company must decide, first, the standard sizes to be stocked (the so called assortment problem) and, second, how to cut the required pieces from the stocked sizes so that the extraneous trim loss is not undesirably high. In this paper, a BOM is assumed to contain a demand (or orders) for rectangular pieces that are to be cut, with minimal trim loss, from larger rectangular stock sheets. The stock sheets are fixed in number and consist of several different types, each type being defined by size in terms of its length and width.

The method that is proposed is based on the following approach: a stock sheet is selected from the available types, and a known layout algorithm is employed to allocate pieces from the BOM to that sheet. When no further pieces can be allocated, a new stock sheet that may or may not be identical to the previous sheet is selected from the available types, and the layout procedure is applied to this new sheet. This continues until the BOM is exhausted. The fact that alternative stock sheet types may be used each time a new sheet is needed means that the order (or sequence) in which the different stock sheets are picked influences the total trim-loss accumulated after the BOM has been fully allocated. Thus, there are multiple BOM. The number of possible sequences depends on the number of different stock sheet types available and the size of the BOM. Even for a small number of stock sheet types, the number of possible sequences that can be investigated is combinatorially explosive.

ISSN 1072-4761

©INTERNATIONAL JOURNAL OF INDUSTRIAL ENGINEERING

A heuristic procedure that aims to reasonably minimize the trim-loss while avoiding the long computational time requirement due to the combinatorial nature of this problem is proposed here. The new heuristic is based on the use of aspiration levels to eliminate unlikely candidate sizes from consideration at each point where a new stock sheet needs to be started. Fair comparisons of different existing procedures with the new heuristic are accomplished by means of tests performed on randomly generated problem sets.

2. LITERATURE REVIEW

There are two distinct approaches for handling the two-dimensional cutting stock problem with multiple stock sizes. In the first, the BOM is allocated among the multiple sheets in a simultaneous manner, while in the second approach the BOM is allocated by packing to completion selected sheets one at a time. Procedures that belong in the category of the first approach include Gilmore and Gomory (1961), who used linear programming to find patterns for the one-dimensional cutting stock problem, and later extended it to two-dimensional problems involving multiple stock sheets (Gilmore and Gomory, 1965). Their method generates optimal results, but it is practical only in small problems due to the near exponential increase in computational demand that arises with each additional BOM piece and each new stock type. Some heuristics in which layouts 'evolve' by adding pieces to a partial layout, such as those proposed by Bengtsson (1982) and Wang (1983), can also be used to simultaneously pack multiple sheets.

However, most research into the two-dimensional cutting stock problem with multiple sheets has favored the second approach, mainly because it circumvents the high complexity associated with simultaneously laying out an entire BOM. This second approach treats the multiple stock sheet problem as two separate sub-problems. The first sub-problem involves laying out pieces from the BOM on any given stock sheet (the single sheet layout problem) selected from the multiple types of available sheets. The second sub-problem is then to decide which sheet types to use, and in what order to process them (the so-called sequencing problem). There are numerous heuristics that have been proposed for packing BOM pieces onto single sheets. In addition to the heuristics of Bengtsson (1982) and Wang (1983), other heuristics range from simple first-fit algorithms (Israni and Sanders, 1979) to algorithms that employ mathematical programming and tree search methods (Adamowicz and Albano, 1976; Chirstofides and Whitlock, 1977; Beasley, 1985; Viswanathan and Bagchi, 1993; and El-Bouri et al., 1994). More recent papers have investigated meta-heuristic methods in packing problems. For example, Burke and Kendall (1999) compared the performances of tabu search, genetic algorithms and simulated annealing in a construction method for building layouts by adding rectangles one at a time to a partial layout; and Leung et al. (2001) combined simulated annealing and genetic algorithms for laying out rectangles into a single sheet. A survey of recent exact and approximation algorithms, as well as heuristic and meta-heuristic methods for packing rectangles on single sheets can be found in Lodi et al. (2002). Furthermore, meta-heuristics have been employed in procedures to assign BOM pieces to multiple sheets simultaneously. Most notably, Babu and Babu (1999) applied a genetic algorithm that uses a first-fit heuristic to allocate pieces according to an order and a selection of sheets that are all specified by means of genetic strings. Onwubolu and Mutingi (2003) also employed genetic algorithms. The application they investigated was that of cutting a list of demanded rectangles from a number of stock sheet rolls of varying widths and infinite lengths.

The sub-problem of selecting stock sheets, and the sequence in which they are processed by a single sheet layout algorithm, is a search type problem that increases in complexity with the number of different stock sheet sizes. The optimal sequence of stock sheets can be determined by enumerative methods, such as by using a tree search with good bounding techniques. A brief overview of papers dealing with stock sheet sequencing is given next, because of their relevance to the heuristic proposed in this paper. Yanasse et al (1991) suggested a heuristic for the trim loss problem which uses a pattern building procedure and an enumeration scheme. First, the BOM pieces are prioritized according to their physical attributes. Then they are assigned, in order of decreasing priority, to the unused space of a stock sheet. When no more order pieces can be assigned to the stock sheet, another priority list is generated and a new cutting pattern is found. This process continues until the BOM is completed. The enumeration scheme generates combinations of stock sheets, ordered in non-decreasing areas, that define the stock sheet size whose cutting pattern is to be determined next. The main drawback of the algorithm is that the computer run-time is probably too large for practical implementation.

Qu and Sanders (1989) introduced a Branch and Bound procedure for finding the optimal sequence of stock sheets. [illegible] solutions [illegible] computational effort may be [illegible] follows any path down the tree. It calls the resulting leaf node the "best" node and stores the cumulative trim loss (CTL) attained at that node, along with the corresponding path and sheet layouts. It then backtracks one level, and tests another branch to complete laying out the same BOM. If the ensuing CTL is less than that of the "best" node, information related to the "best" node is updated. The procedure is repeated until all paths are exhausted. The STEP procedure, on the other hand, selects the node having the least CTL among the N nodes at each level, and discards the rest of the nodes. Branching from this node, the operation is repeated on the next level, and so on until the BOM is completed. Therefore, the computational time is significantly reduced in the STEP procedure, but often at the expense of a higher trim loss than can be obtained by using the PREORDER procedure.

The approach suggested in this paper proposes to improve the quality of the solutions obtained by the STEP procedure, while maintaining reasonably close computational requirements. The strategy is to avoid the extreme local minima that are characteristic of the STEP procedure, so that the tree search is guided by a more global target. Furthermore, the papers reviewed above for the sequence selection problem used simple first-fit heuristics for laying out BOM pieces on the single sheets. The alternative procedure that is chosen here for laying out the BOM pieces on individual stock sheets, known as the Single Sheet Layout (SSL) heuristic, invariably follows the rule-based approach of El-Bouri et al (1994), which was shown to be quite efficient for a single stock sheet size.

3. METHODOLOGY

The procedure proposed for the two-dimensional layout of pieces on a selection of stock sheets (from multiple available sizes) is called the Trim-Loss Reduction Method Based on Aspiration Level (TRIMBAL). TRIMBAL introduces the idea of a Basic Stock Sheet size and an aspiration level to control the number of stock sheet sequences that need to be considered. The method aims to improve the performance of the STEP heuristic (Qu and Sanders, 1989) with minimal additional computational demand.

Consider N different sizes (types) of stock sheet. The number available for each type is assumed to be unlimited. Let

L_k = length of type k stock sheet.
W_k = width of type k stock sheet.
LTL_i = (local) trim-loss in the i^{th} sheet of a sequence of selected stock sheets.
$LTL_{i,k}$ = (local) trim-loss when sheet type k is used for the i^{th} sheet in the sequence under consideration.
TTL = total trim loss.
TTL_k = total trim loss when the SSL algorithm is applied and only type k stock sheets are used.
M = total number of sheets needed to satisfy the BOM.
M_k = total number of sheets needed to satisfy BOM when only type k stock sheets are used.
S_i = stock sheet type occupying the i^{th} position in a sequence.

The value of M is sequence-dependent. The greater is the proportion of smaller stock sheet sizes used, the larger is the likely value of M. Furthermore, the trim-loss on the last sheet is not counted. This is so that the packing density can be better compared between different sequences. Hence, the total trim-loss for a given sequence is given by

$$TTL = \sum_{i=1}^{M-1} LTL_i \qquad \dots \qquad (1)$$

Assuming that only one of the N stock sheet types may be used, the type that generates the least total trim-loss for the BOM is called a Basic Stock Sheet (BSS). In order to determine which sheet type is the BSS for a given BOM, the SSL algorithm is applied to lay out the BOM N separate times, each time using one of the available sheet types. This method for specifying the BSS size is called the Least Trim-Loss Method (LTM). The LTM has the disadvantage of computational burden when the BOM is very long, or when the number of available stock sheet types is great.

The TRIMBAL procedure starts with an initial sequence created by allocating the BOM using only the stock sheet type that has been identified as the BSS. Although a solution that uses only BSS types may give the lowest cumulative trim loss, the lowest possible local trim loss on any individual sheet is not necessarily achieved by a BSS. In other words, there might be another stock sheet size that can produce a lower local trim loss if it were to replace the BSS in the i^{th} sheet in the sequence.

The search for improved sequences is performed by deciding whether any of the Basic Stock Sheets can be replaced by other sizes. These decisions are based on a user-defined aspiration level for the local trim-loss in each replacement sheet for the BSS. If, at any stage i, the BSS does not meet the aspiration level, other stock sheet sizes are examined to find, if the aspiration level is set too low, all or most of the BSS are subject to replacement, with a likely result of increased computational load. On the other hand, replacement of the BSS becomes rare if the aspiration level is set too high and the problem reverts to one in which all or most of the stock sheets are a BSS type. Consequently, the flexibility of having different stock sheet sizes is lost. A more balanced aspiration level can be obtained from the average of the local trim losses

in all of the sheets in the BSS sequence. Once the BSS and the aspiration level (AL) have been determined, they are altered only when a new BOM or a new set of stock sheet sizes is considered.
Details of the TRIMBAL heuristic are outlined in the following series of steps.

<u>Step 1:</u>

1.1 Sort the stock sheet types in order of non-increasing size.
1.2 Set $k = 1$.
1.3 Lay out the entire BOM using the k^{th} stock sheet type, and calculate TTL_k .
1.4 If $k < N$, increment k by 1 and return to step 1.3. Otherwise let k^* be the stock sheet type that gives the minimum TTL_k. for $k = 1, \ldots N$.
1.5 Designate stock sheet type k^* as the BSS.
1.6 Set aspiration level $AL = TTL_{k^*}/(M_{k^*} - 1)$.

<u>Step 2</u>

2.1 Generate initial sequence $S_i = k^*$ for $i = 1, \ldots M_{k^*}$.
2.2 Set stage $i = 1$,
2.3 Apply SSL algorithm for $S_i = k^*$.
2.4 If $LTL_i \leq AL$, proceed to step 2.5.
If $LTL_i > AL$, apply SSL algorithm to the other (N-1) stock sheet types, starting from the top of the ordered list of stock sheets. Let k' be the first stock sheet found such that $LTL_{i,k'} < AL$.
  - If k' does not exist, then S_i equals the stock sheet type that produces the minimum $LTL_{i,k}$ for $k = 1, \ldots i$.
  - Otherwise, $S_i = LTL_{i,k'}$
2.5 Update the BOM by removing all pieces allocated to stock sheet S_i. If the BOM is empty, then STOP and calculate TTL. Otherwise, set $i = i + 1$ and return to step 2.3.

3.1 Illustrative Example

The TRIMBAL procedure is demonstrated by means of a numerical example. A BOM consisting of five piece types is given in Table 1. There are three different stock sheet types available, the sizes of which are listed in the last column of Table 1. The calculations in step 1 of the algorithm that result in the determination of the BSS are summarized in Figure 1. Figure 1 shows the local and total trim loss for each stock sheet that arises from the application of the SSL algorithm for each of the available stock sheet types. Stock sheet type 2 produces the lowest total trim loss (2844), and it is, therefore, the BSS.

Table 1. Example BOM with three stock sheet types

Piece	Length	Width	Quantity	Stock Sheet Type	Dimensions ($L_i * W_i$)
A	45	21	8	1	149 x 51
B	43	14	7	2	140 x 44
C	42	29	9	3	130 x 50
D	38	26	1		
E	19	10	5		

The aspiration level is determined from the mean of the local trim losses on the four BSS sheets, as considered in Figure 1. Therefore, $AL = (320+320+1714+490)/4 = 711$. The search part of the algorithm (*Step 2*) selects a sheet at each stage in the following manner. (Note that $LTL_{i,k}$ is always obtained by the application of the SSL heuristic to sheet k at stage i.)

<u>Stage 1</u>

For BSS (sheet 2), $LTL_{1,2} = 320 <$ AL. Therefore BSS is retained for stage 1.
The SSL algorithm allocates the following pieces from the BOM (quantity x type) when sheet type 2 is used at stage 1: (3 x B), (3 x C), and (2 x E).

Stage 2, $LTL_{2,2}$ = 320 < AL. Therefore BSS is retained for stage 2.
SSL algorithm allocates the following pieces when sheet type 2 is used in stage 2: (3 x B), (3 x C) and (2 x E).

<u>Stage 3</u>

Updated BOM: (8 x A), (1 x B), (3 x C), (1 x D) and (1 x E).
For BSS, $LTL_{3,2} = 1714 > AL$. Therefore BSS is subject to replacement.
For sheet type 1, $LTL_{3,1} = 984 > AL$, and for sheet type 2, $LTL_{3,3} = 766 > AL$.

None of the layouts meets the aspiration level. The one with the minimum trim loss, sheet type 3, is chosen.
SSL algorithm allocates the following BOM pieces when sheet type 3 is used at stage 3: (2 x A), (3 x C), and (1 x E).

Stage 4

Updated BOM: (6 x A), (1 x B) and (1 x D).
For BSS, $LTL_{4,2} = 790 > AL$. Therefore BSS is subject to replacement.
For sheet type 1, $LTL_{4,1} = 1929 > AL$, and for sheet type 3, $TL_{4,3} = 830 > AL$.
None of the layouts meets the aspiration level. The one with the minimum trim loss, sheet type 2 (the BSS), is chosen.
SSL algorithm allocates the following pieces when sheet type 2 is used at stage 4: (4 x A), (1 x B) and (1 x D).

After stage 4, only two pieces of part type A remain and they are allocated to the smallest sheet. This last sheet is not considered fully packed, so that it is not accounted for in the total trim loss calculations. The solution provided by TRIMBAL is therefore : $S_1 = 2$; $S_2 = 2$; $S_3 = 3$; $S_4 = 2$; and $TTL = 2196$. This means that the SSL algorithm is to be applied to sheet types 2, 2, 3, and then 2 again, in that order.

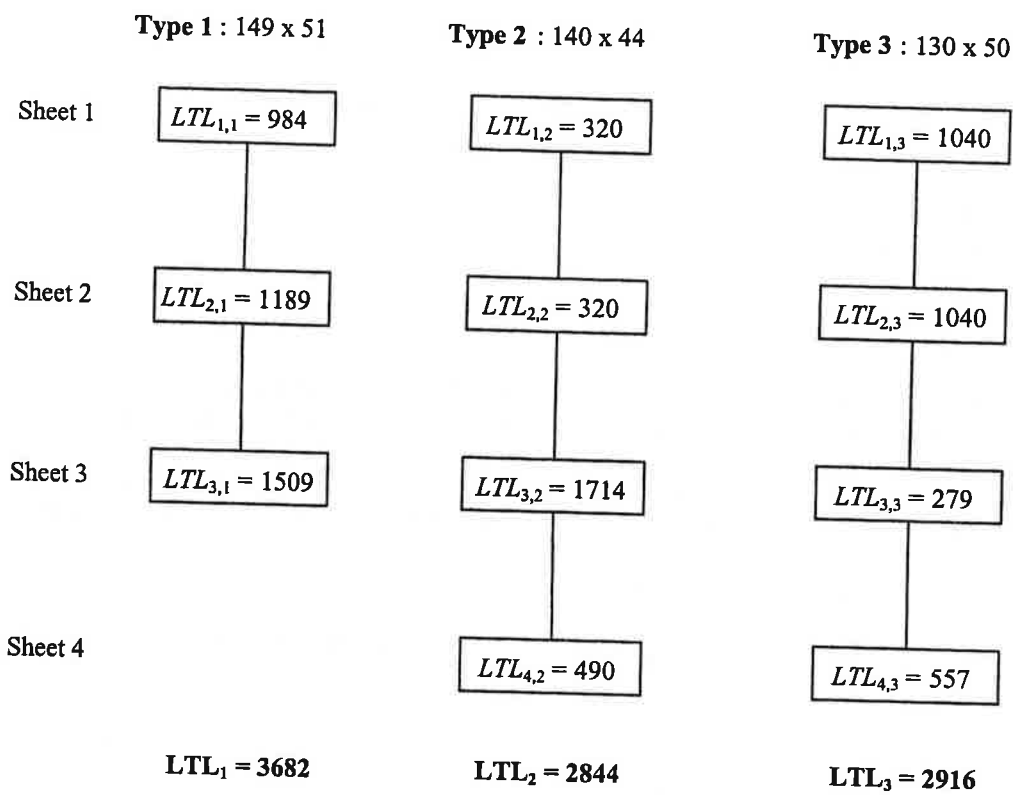

Figure 1: Trim losses resulting from application of SSLalgorithm for each of the available stock sheet types

4. COMPUTATIONAL EXPERIENCE

A disadvantage for TRIMBAL when dealing with combinations of large BOMS and numerous stock sheet types is the computational effort demanded by the enumerative strategy of the LTM for finding the BSS type. An alternative to LTM that significantly reduces this effort, is to simply use the stock sheet that has the largest area, and a reasonably low length-to-width ratio (i.e aspect ratio). This alternative strategy is called the Largest Area Method (LAM). It is based on a general observation that larger stock sheets tend to produce less trim-loss than smaller sheets. However, a stock sheet's aspect ratio whose aspect ratio is also greater than a (somewhat arbitrarily selected) value of 5.0.

TRIMBAL is compared here with the Branch and Bound (B-B), PREORDER and STEP procedures of Qu and Sanders (1989) for five problem categories. All four procedures were programmed and tested using Turbo Prolog by Borland International. A problem category is defined by the ratio of the mean area of the demanded pieces, $\bar{A}_O$, to the mean area of one sheet of each of the different sizes stocked, $\bar{A}_S$. The ratio $\bar{A}_O / \bar{A}_S$, called APSA for convenience, does not depend upon the available quantity of each stock sheet size. The categories considered have APSA ratios corresponding to 0.04, 0.10, 0.25, 0.50 and 1.00. Test problems in each of the five cagegories are further divided into subcategories, based on the available number of different stock sheet types. Test problems may have between two and six different stock sheet types so that five subcategories are created (one for each number of stock sheet type). Comparisons are performed for the resulting 5 x 5 = 25 cases by randomly generating 30 test problems in each subcategory. Thus, a total of 750 problems is solved by all the procedures, with the exception of B-B. Due to the limited computer memory and time resources, the B-B procedure is employed only when a problem has three or fewer different stock sheet types. This means that B-B is considered in only 300 (5 x 2 x 30) of the test problems.

A problem is formulated by randomly creating a BOM of demanded rectangles that meet specified dimensions and dimensional ratios. The methodology is similar to that described by Bischoff and Ratclife (1995), but for a rectangle rather than a three dimensional container. Stock sheets for a given APSA ratio are generated with random lengths and widths between an arbitrary 25 to 120 units, and with a standard deviation in area that is between $0.1(\overline{A}_S)$ and $0.5(\overline{A}_S)$. Stock sheet widths are invariably larger than the maximum length of a BOM piece. The number of different piece types in a BOM ranges from 8 to 50, and the demanded quantity for any one piece does not exceed 20. The total number of order pieces to fill a BOM varies from 10 to 205. Details are given by Rao (1997), who also found empirically that LAM is computationally preferable to LTM in determining the BSS when APSA ratios are equal to or less than 0.1 and the number of available stock sheet types is three or more. Therefore, TRIMBAL employs LAM exclusively in these cases, and LTM otherwise.

5. RESULTS

Figures 2 and 3 compare the four procedures based upon the Mean (Percentage) Stock Sheet Utilization, MSSU, and the mean CPU run times. The percentage utilization of an individual stock sheet is 100% times the total area of the order pieces allocated to it, divided by the stock sheet's area. The MSSU is simply the mean of the percentage utilizations over all the stock sheets used to fulfil a BOM. The mean of the CPU times, on the other hand, is taken over a stated number of stock sheet sizes in a particular APSA category. The figures indicate that, regardless of the procedure adopted, the MSSU and corresponding mean CPU time generally increase as the APSA ratio decreases and, for a given APSA ratio, as the number of stock sheet types increases. Not unexpectedly, the B-B procedure always gives the highest MSSU in Figure 2 followed by PREORDER, TRIMBAL and STEP. Differences in the MSSU between the last three procedures decrease progressively as the APSA ratio is reduced. They become almost negligible at the two smallest APSA ratios when the number of stock sheet types increases above four. Clearly, TRIMBAL is seen from Figure 2 to generate a MSSU that is higher than STEP, but not as high as that produced by PREORDER. A two-sample *t*-test comparison at 95% level of confidence indicated that the difference in MSSU between TRIMBAL and STEP was statistically significant in all categories having APSA ratios of 0.25 and above, and in 80% of the subcategories with an APSA ratio of 0.10. Conversely, the *t*-test failed to detect a significant difference between the two procedures in 80% of the subcategories having an APSA ratio of 0.04. A similar *t*-test comparison for the MSSU between TRIMBAL and PREORDER indicates a significant difference in two out of the five subcategories tested in each of the categories with APSA ratios of 1.00, 0.50 and 0.10, and in only one out of the five subcategories tested for APSA ratio of 0.25. The *t*-test results indicate that the performance of TRIMBAL, in terms of MSSU, is much closer to PREORDER, and significantly better than STEP.

Concerning the computational time, the mean run times needed when using TRIMBAL are seen to be only slightly higher than those consumed by STEP when applied to the same test problems. In fact, a *t*-test at a 95% confidence level fails to confirm a significant difference in the mean run times between the two methods in about half the cases tested. Furthermore, the computational times for both the TRIMBAL and STEP procedures are almost always noticeably less than those of PREORDER. They are smaller still when compared with the time required by the B-B procedure to find the optimal sequences. In addition, of course, B-B has the most computer storage needs. Therefore a procedural choice depends upon a perceived balance between the conflicting demands of better stock sheet utilization and additional computer requirements. As the number of stock sheet sizes rises above two, the additional computational penalty imposed by B-B and PREORDER becomes increasingly less justified by the corresponding increase in stock utilization. This observation is particularly true when the APSA ratio falls below 0.50. Then, the consistently superior stock utilization of TRIMBAL warrants its slightly loss values in the BSS from the AL. The greater is this deviation, the less is the expected sensitivity to variations in AL. In the problems tested in this study, the minimum observed deviations between local trim losses and the AL was typically of

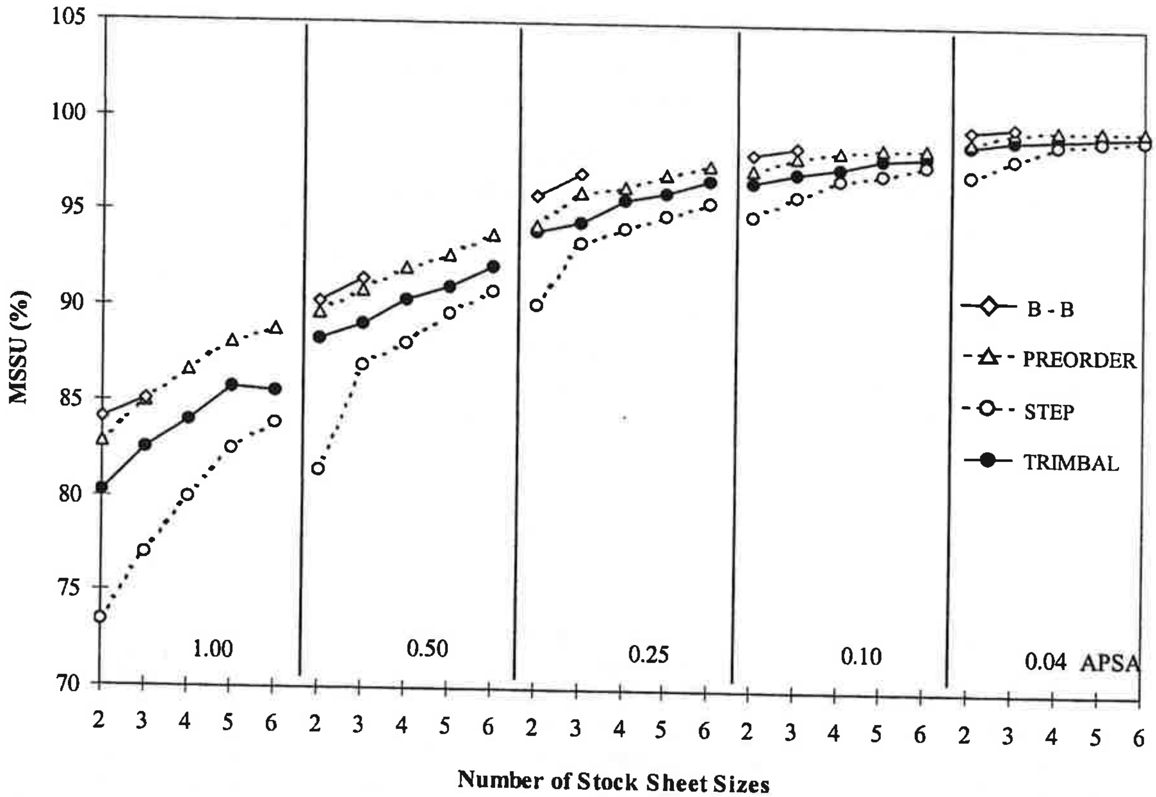

Figure 2 : The mean percentage stock sheet utilization (MSSU) for TRIMBAL in comparison with three other heuristics.

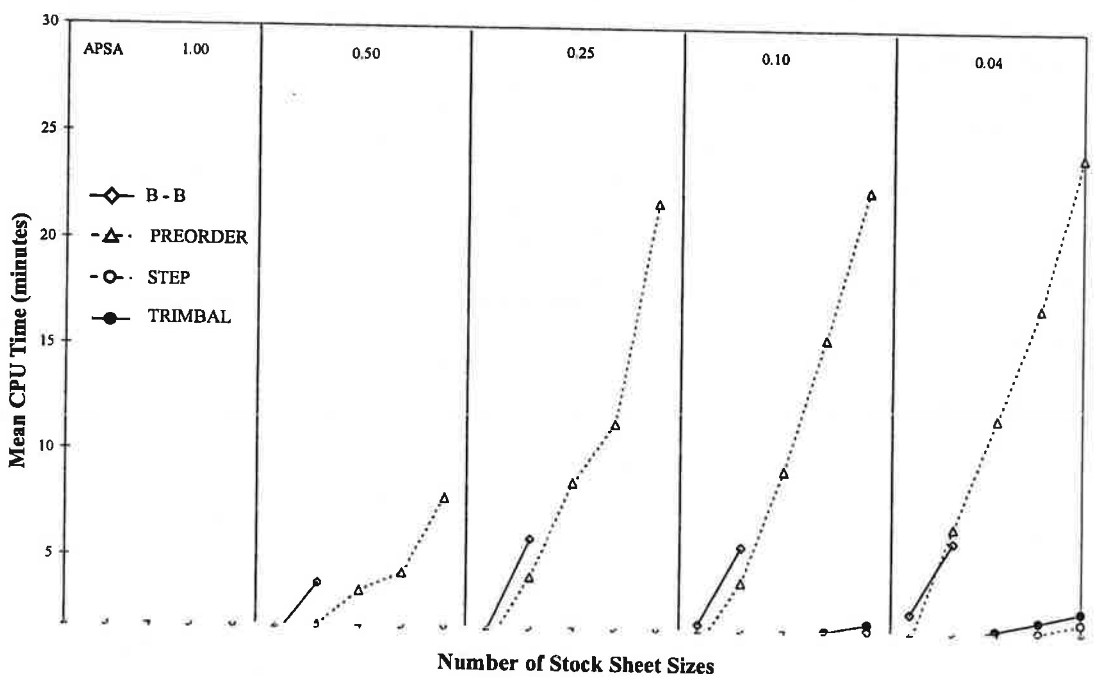

Figure 3: The mean CPU times consumed by TRIMBAL in comparison with three other heuristics.

the order of a few percentage points, if not more. Thus, TRIMBAL might be expected in most cases to be relatively insensitive to variations of a few percentage points (5% or less) in AL. To investigate further, a small sample of randomly generated test problems was examined in detail to determine the effect of changes of 10% and 25% in the AL. The results indicated that variations up to ±10% in AL did not affect the final results in 4 out of 10 instances. On the other hand, only 1 out 10 cases remained unaffected by variations within ±25% of the AL. The implication is that if it is desired that TRIMBAL expand its search of non BSS sizes, then the AL should be reduced by around 25% or more, and vice versa.

6. CONCLUSIONS

This paper described a new heuristic (TRIMBAL) for minimizing the trim loss in cutting stock problems when multiple stock sheet types are available for laying out a BOM. The heuristic improves upon two of the methods (STEP and PREORDER) proposed by Qu and Sanders (1989), by introducing an aspiration level to guide the search for a near-optimal sequence of stock sheet selections. Results obtained from test problems covering a fair range of stock sheet types and BOM sizes show that the new heuristic fares significantly better in minimizing the trim-loss than does the STEP procedure, while consuming a comparable computational time. Like most heuristics, however, TRIMBAL's main limitation is that it does not guarantee an optimal solution. Therefore, future research efforts involving TRIMBAL should be preferably directed towards further improving the quality of its solutions without compromising computational time. The layout algorithm that TRIMBAL employs to allocate pieces to a sheet typically produces lower trim losses in the initial sheets than it does later on as the BOM is gradually exhausted. Using an AL based on average trim loss is useful in this case because it allows the heuristic to investigate a greater number of alternative stock sheet sizes towards the latter stages of the solution. Nevertheless, there still remain opportunities for researching more sophisticated means and tools for selecting a more effective AL. One possible approach, for instance, is to design an adaptable AL that adjusts itself dynamically as the composition of the BOM changes during the layout procedure.

ACKNOWLEDGMENTS

The financial support of the National Sciences and Engineering Research Council of Canada is acknowledged gratefully.

7. REFERENCES

1. Adamowicz, M., and Albano, A., (1976). A solution of the rectangular cutting stock problem. IEEE Transactions on Systems, Man and Cybernetics, SMC-6, 302-310.
2. Babu, A.R., and Babu, N.R., (1999). Effective nesting of rectangular parts in multiple rectangular sheets using genetic and heuristic algorithms. International Journal of Production Research, 37(7): 1625-1643.
3. Beasley, J.E., (1985). An exact two-dimensional non-guillotine cutting tree-search procedure. Operations Research, 33: 49-64.
4. Bengtsson, B.E., (1982). Packing rectangular pieces - a heuristic approach. The Computer Journal, 25: 353-357.
5. Bischoff, E., and Ratclife, M., (1995). Issues in the development of approaches to container loading. International Journal of Management Science, 37: 383-386.
6. Burke, E., and Kendall, G., (1999). Comparison of meta-heuristic algorithms for clustering rectangles. Computers & Industrial Engineering, 23(4): 377-390.
7. Christofides, N. and Whitlock, C., (1977). An algorithm for two dimensional cutting problems. Operations Research, 25: 30-44.
8. El-Bouri, A., Popplewell, N., Balakrishnan, S., and Alfa, A., (1994). A search-based heuristic for the two-dimensional bin-packing problem. INFOR, 32(4): 265-274.
9. Gilmore, P.C., and Gomory R.E., (1961). A linear programming approach to the cutting stock problem. Operations Research, 9: 849-859.
11. Leung, T.W., Yung, C.H., and Troutt, M.D., (2001). Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem. Computers & Industrial Engineering, 40: 201-214.
12. Lodi, A., Martello, S., and Vigo, D., (2002). Recent advances on two-dimensional bin packing prolems. Discrete Applied Mathematics, 123: 379-396.

13. Onwubolu, G.C., and Mutingi M., (2003). A genetic algorithm approach for the cutting stock problem. Journal of Intelligent Manufacturing, 14: 209-218.
14. Qu, W., and Sanders, J.L., (1989). Sequence selection of stock sheets in two-dimensional layout problems. International Journal of Production Research, 27(9): 1553-1571.
15. Rao, J., Sequencing Orders of Multiple Sized Stock Sheets. M.Sc. Thesis, University of Manitoba, 1997.
16. Viswanathan, K.V., and Bagchi, A., (1993). Best-first search methods for constrained two-dimensional cutting stock problems. Operations Research, 41(4): 768-776.
17. Wang, P.Y., (1983). Two algorithms for constrained two-dimensional cutting stock problems. Operations Research, 31: 573-586.
18. Yanasse, H.H., Zinober, A.S.I., and Harris, R.G., (1991), Two-dimensional cutting stock with multiple stock sizes. Journal of the Operational Research Society, 42(8): 673-683.