Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management
Bali, Indonesia, January 7 – 9, 2014

An Algorithm for Solving Lot Sizing and Cutting Stock Problem within Aluminum Fabrication Industry

S. M. A. Suliman
Mechanical Engineering Department, University of Bahrain,
P. O. Box 32038, Isa Town, Bahrain

Abstract

Needless to say that the combined lot sizing and cutting stock problem is very relevant in a wide range of manufacturing industries. This paper presents an algorithmic solution approach to overcome the difficulty in solving non-linear integer formulation of the problem. The algorithm is based on the traditional approach where the lot sizing is determined for each period, and then the best cutting patterns are generated. However the algorithm adopts a backward lot-sizing based on available capacity and purchase cost burden index. Using 85 instances, the performance of the algorithm is assessed in terms of trim loss and computation time for a range of different piece lengths; number of final products; and number of scheduling periods. Good results are produced when compared with the practiced performance.

Keywords:
Cutting stock problem; lot sizing; nonlinear integer programming; manufacturing; aluminum fabrication.

1. Introduction

Early efforts that can be considered of integral nature between LS and CSP were done by Haessler [1], and deCarvalho and Rodriguez [2] where they considered both the cost of setups and the cost of trim loss. Explicit handling of the combined problem is reported by Krichagina et al [3] from paper industry and Hendry et al [4] from copper industry. Two-stage solution procedures are presented in which the CSP is solved in the first stage, while the LSP is solved in the second stage. Nonas and Thorstenson [5] formulated the combined problem with a joint setup cost and integer lot sizes resulting into a non-linear minimization problem with a concave objective function and linear constraints. Different solution algorithms are presented and categorized as global or local algorithms. Their global algorithms are based on the extreme point ranking method of Murty [6]. They provided optimal results for problems having up to 50 different cutting patterns. The second category of the algorithms is based on three local search methods that operate around the optimal cutting stock solution. The number of cutting patterns that can be generated for the CSP limits the use of these algorithms.

Arbib and Marinelli [7] proposed an integer programming model integrating decisions at both planning (LSP) and the operational (CSP) levels, with the aim of minimizing production, holding, and transportation costs. The model is examined on a real two-stage flow line where components are produced at a first stage cutting process with skiving option followed by the downstream assembly stage. Gramani and Franca [8] formulated a mathematical model for the combined problem, and solved it heuristically using a network shortest path problem representation. In this representation, each arc of the network is associated with a capacitated CSP that is solved by the classical simplex algorithm with Gilmore and Gomory [9] column generation procedure. However, their contribution is limited to the pieces level and does not consider the final products.

A set of solutions are generated iteratively in a tree structure using a column generating solution procedure (a sequential cutting-stock tree-search heuristic (TSH)) that considers both the trim loss and the number of different cutting patterns [5]. Nonas and Thorstenson [10] suggested a new column generating solution procedure that improves the performance of the heuristic on both small and large-sized instances of the combined problem. The proposed procedure includes characteristics from both Haessler's sequential procedure and Nonas and Thorsternson's TSH [1, 5]. Gramani et al [11] presented a mathematical model for the combined problem. They proposed for their model a resolution approach based on lagrangian relaxation, and using the sub-gradient method to update the Lagrange multiplier vectors.

Recently, Erjavee et al [12] presented a method for assessing the optimal stock size for the expected order size for a single-period one-dimensional CSP. They proved that the stock size is optimal when the expected total costs of trim loss, warehousing, and non-fulfillment are optimal. The few reported research endeavors on the combined lot sizing problem and cutting stock problem either they do not describe the objectives and constraints sufficiently or their solution approaches are inadequate to provide reasonable performance quality- and time-wise.

2. Problem Formulation

Consider an aluminium extrusion sections of standard length L to be cut into P pieces ordered lengths l_i $(i = 1,2, \dots, P)$. The demand for these pieces in period t depends on lot size decision of how many final products to be manufactured in the period. The pieces can be produced in-house or bought-out. To formulate the combined cutting stock and lot sizing problem, let T be the number of periods in the planning horizon, M the number of different types of final products demanded, P the number of different piece types (lengths), a_{ij} the number of units of piece i in cutting pattern j, and N the number of all possible cutting patterns.

The problem addresses the issues of how to establish the cutting patterns of the pieces $a_{ij}(i = 1, 2, \dots, P; j = 1, 2, \dots, N)$, in case of make decision, and lot sizes for final products with the objective of minimizing the total cost over the planning horizon. The total cost C_T in our case is composed of production cost C_{pro} (material, processing, and set-up cost), purchasing cost C_{pur}, inventory cost C_{inv}, and order non-fulfillment cost C_{nful}. These costs can be related as follows:

$$C_T = C_{pro} + C_{pur} + C_{inv} + C_{nful}$$

The production cost consists of material, set-up, and processing costs which can be expressed by:

$$C_{pro} = \sum_{t=1}^{T} \left\{ c_m L \sum_{j=1}^{N} s_j^t + \sum_{j=1}^{N} c_{sj} y_{sj}^t + \sum_{j=1}^{N} \sum_{i=1}^{P} c_{pi}\, a_{ij}\, s_j^t \right\} \quad (1)$$

Where c_m is the unit material cost, c_{sj} is the set-up cost for pattern j, c_{pi} is the processing cost per unit of piece i, r_{ik} is the number of units of piece i to produce one unit of final product k.

The decision variables s_j^t and x_k^t are the number of aluminium sections cut according to pattern j in period t, and the lot size of the final product k produced in period t, respectively. y_{sj}^t is a binary variable defined as:

$$y_{sj}^t = 1, if\ s_j^t > 0$$
$$= 0, if\ s_j^t = 0 \quad (2)$$

The three parts of expression (1) above represent material, set-up, and processing costs, respectively.
For purchased pieces, the cost can be defined by:

$$C_{pur} = \sum_{t=1}^{T} \sum_{i=1}^{P} c_{bi} \{ \sum_{k=1}^{M} r_{ik} x_k^t - \sum_{j=1}^{N} a_{ij} s_j^t \} \quad (3)$$

Where c_{bi} is the purchasing cost per unit of piece i.
Three types of inventories should be considered in the cost function: material, pieces, and final product inventories.

$$C_{inv} = \sum_{t=1}^{T} \{ (S^t - \sum_{j=1}^{N} s_j^t) h_m + \sum_{i=1}^{P} (I_{pi}^t + \sum_{j=1}^{N} a_{ij} s_j^t + (\sum_{k=1}^{M} r_{ik} x_k^t - \sum_{j=1}^{N} a_{ij} s_j^t)(1 - y_{pi}^t) - \sum_{k=1}^{M} r_{ik} d_k^t) h_{pi} + \sum_{k=1}^{M} (I_{fk}^t + x_k^t - d_k^t) y_{fk}^t h_{fk} \} \quad (4)$$

Where h_m is the unit material inventory cost, h_{pi} is the inventory cost per unit of piece i, h_{fk} is the inventory cost per unit of final product k. S^t is the amount available of aluminium sections of standard length in period t. d_k^t is the demand of final product k in period t. a_{ij} is the number of units of piece i cut according to pattern j. I_{pi}^t and I_{fk}^t are the inventories of piece i and final product k in period t, respectively. y_{pi}^t and y_{fk}^t are binary variables defined as:

$$y_{pi}^t = 1,\ if\ \sum_{j=1}^{N} a_{ij} s_j^t \geq \sum_{k=1}^{M} r_{ik} x_t^t \text{ (pieces demand is satisfied)}$$
$$= 0,\ otherwise \quad \text{(pieces demand is satisfied partially by production}$$

and partially by purchasing). (5)

And

$$y_{fk}^t = 1, if\ I_{fk}^t + x_k^t - d_k^t \geq 0,\ \textit{i.e. fulfilled demand}$$
$$= 0, \textit{otherwise, i.e. non-fulfilled demand} \quad (6)$$

The final component of the total cost is the demand non-fulfillment cost, which is

$$C_{nful} = \sum_{t=1}^{T} \sum_{k=1}^{M} \left[d_k^t - \left(I_{fk}^t + x_k^t\right)\right]\left(1 - y_{fk}^t\right) c_{nk} \quad (7)$$

Where c_{nk} is the cost of non-fulfillment demand of final product k.
The following constraints should be considered.

Demand and inventories: The relationship between the demand of the final products $d_k^t (k = 1,2, \dots, M)$ for each planning period t to be satisfied and the associated inventory is given by:

$$(I_{fk}^t + x_k^t - d_k^t) y_{fk}^t = I_{fk}^{t+1},\ \text{for all } k \text{ and } t \quad (8)$$

Consequently, the piece inventories can be defined as:

$$I_{pi}^t + \sum_{j=1}^{N} a_{ij}\, s_j^t\ - \sum_{k=1}^{M} r_{ik}\, d_k^t = I_i^{t+1},\ for\ all\ i\ and\ t \quad (9)$$

The initial inventories (i.e. at beginning of period 1) of final products as well as of the pieces can be considered null, i.e.

$$I_{fk}^1 = I_i^1 = 0 \quad (10)$$

Material availability: To ensure aluminium sections availability, the following relationship should hold:

$$S^t - \sum_{j=1}^{N} s_j^t \geq 0\,, for\ all\ t \quad (11)$$

Another material constraint is related to the standard length of the aluminium sections:

$$\sum_{i=1}^{P} a_{ij}\, l_i\ \leq L,\ for\ all\ j\ and\ t \quad (12)$$

Capacity: The processing capacity is given in terms of total number of cuts that can be made in period t, and designated b^t, thus, the capacity constraint can be defined by:

$$\sum_{j=1}^{N} \sum_{i=1}^{P} a_{ij}\, s_j^t \leq b^t, for\ all\ t \quad (13)$$

The non-linear-integer mathematical model of the combined cutting stock and lot sizing problem can be written then as follows:

$$Min\ C_T = \left(C_{pro} + C_{pur} + C_{inv} + C_{nful}\right) \quad (14)$$

Subject to the constraints of expressions (2), (5), (6), (8) to (13) in addition to the following non-negativity constraints:

$$x_k^t\ \geq 0, and\ integer\ for\ all\ k\ and\ t$$
$$s_j^t \geq 0, and\ integer\ for\ all\ j\ and\ t$$
$$I_{pi}^t \geq 0, and\ integer\ for\ all\ i\ and\ t$$
$$I_{fk}^t \geq 0, and\ integer\ for\ all\ k\ and\ t$$
$$y_{sj}^t = \{0,1\}, for\ all\ j\ and\ t.$$
$$y_i^t = \{0,1\}, for\ all\ i\ and\ t.$$
$$y_{fk}^t = \{0,1\}, for\ all\ k\ and\ t. \quad (15)$$

3. Solution Algorithm

The non-linear integer program (NLIP) model described in Section 2 above for the combined LS and CSP is difficult to solve especially for problems that have large number of variables. To overcome this difficulty, an algorithmic solution method (LS-CSP algorithm) is presented. The LS-CSP algorithm is based on a backward lot sizing solution approach. It starts with the last period in the planning horizon (period number T) for which the quantities of the final products to be manufactured (x_k^T) are established based on the period demand (d_k^T) and the available capacity(b^T). If the available capacity is less than the required capacity to satisfy period demand, then the assumption is that period capacity shortage will be compensated from inventories. This compensation approach is the best decision as long as the total cost of made in-house inventories is less than the cost of purchased products. In case of capacity shortage,

selection of the products to be manufactured in the current period is based on a purchasing cost burden index δ^k defined as:

$$\delta^k = \frac{c^k_{pur} - c^k_{prod}}{\sum_{\forall i} r_{ik}} \quad (16)$$

Where

$$c^k_{prod} = c_m \sum_{\forall i} r_{ik} * \ell_i + \sum_{\forall i} c_{pi} * r_{ik}$$
$$= \sum_{\forall i} r_{ik}\left(c_m \ell_i + c_{pi}\right)$$
$$C^k_{pur} = \sum_{\forall i} c_{bi} * r_{ik}$$

Having selected products to be manufactured and their quantities for the last period, the next step is to decide on the cutting patterns. This is done by a pattern generation selection algorithm (Pat-Gen-Sel algorithm).

The LS-CSP algorithm proceeds in the backward direction from the last planning period. It establishes for each period the products to be produced, their quantities, and the cutting patterns to be used. This is done under the constraints of current demand, capacity shortage from previous step, and available capacity.
Along all these steps, it is assumed that inventory is always less expensive than purchasing cost. When the algorithm reaches the first planning period a decision should be made in the case of first period capacity shortage about whether to satisfy this shortage by purchasing or to consider it as non-fulfilled demand. The decision will be based on cost basis.
The LS-CSP algorithm proceeds along the following steps:

1. Establish capacity requirement to satisfy the demand of each period ($\mathcal{b}_r^t$):

$$\mathcal{b}_r^t = \sum_{\forall k} \sum_{\forall i} r_{ik} * d_k^t, \text{ for all } t$$

2. Set the maximum number of pattern selection criteria index to the required level, and move backward from planning period *T* towards period *1*, starting with period *T:*

2.1 If $\mathcal{b}_r^T \leq b^T$, then d_k^T (*k=1,2,…,M*) can be satisfied,
i.e.:

$$x_k^T = d_k^T, \text{ for all } k \text{ and } \Delta b^T = 0$$

where Δb^T is the capacity shortage.
Call Pat-Gen-Sel algorithm.

2.2 If $\mathcal{b}_r^T > b^T$, then capacity shortage Δb^T is:

$$\Delta b^T = b_r^T - b^T$$

This will be satisfied by inventories of earlier periods.

- Prioritize final products *k (k=1,2,…,M)* in-descending order of:
  - purchasing burden index δ^k, or
  - non-fulfillment cost.

Select products with highest rank for production in the current period until the available capacity b^T is exhausted. Thus, products to be produced x_k^T are determined as well as the unsatisfied ones for the current period i.e. $d_{k'}^T$.
Call Pat-Gen-Sel algorithm.

3. For period *t (t = T-1, T-2, …, 1)* do the following:

3.1 If $b_r^t \leq b^t$ and $\Delta b^{t+1} = 0$, then
$x_k^t = d_k^t$, (*k=1,2,…,M*) and $\Delta b^t = 0$.
Call Pat-Gen-Sel algorithm.

3.2 For $b_r^t < b^t$ and $\Delta b^{t+1} \geq 0$.

(a) If $b_r^t + \Delta b^{t+1} \leq b^t$, then the demand of the current period as well as <u>all</u> unsatisfied demand in period *t+1* will be satisfied. Thus,
$x_k^t = d_k^t + I_k^t$, (*k=1,2,…,M*)
where:
$I_k^t = d_k^{t+1}$ = unsatisfied demand in period *t+1*.
As a result no shortage in capacity is manifested in the current period, i.e. $\Delta b^t = 0$.
Call Pat-Gen-Sel algorithm

(b) If $b_r^t + \Delta b^{t+1} > b^t$, then the demand of the current period $(d_k^t, k = 1,2, \dots, M)$ and part of the unsatisfied demand in period $t+1$ $(d_{k'}^{t+1})$ will be satisfied.
Products of unsatisfied demand $d_{k'}^{t+1}$ are ranked in descending order of $\delta^{k'}$ or non-fulfillment cost. Products with the highest ranks are added to the lot size of the current period, i.e.:
$x_k^t = d_k^t + I_k^t$
where $I_k^t = d_{k'}^{t+1} - d_{k'}^t$
$d_{k'}^t$ is the remaining unsatisfied demand at current period t from $d_{k'}^{t+1}$ of period $t+1$ resulting into a capacity shortage of:
$\Delta b^t = b_r^t + \Delta b^{t+1} - b^t$.
Call Pat-Gen-Sel algorithm.

3.3 For $b_r^t > b^t$ and $\Delta b^{t+1} \geq O$.
The demand of the current period t will be partially satisfied.
Rank products k ($k=1, 2, \dots, M$) in descending order of δ^k or non-fulfillment cost. Products with highest ranks are selected for the lot size of the current period until the available capacity b^t is exhausted. Thus, x_k^t is determined leaving $d_{k'}^t$ unsatisfied for the current period.
$I_k^t = 0, and\ \Delta b^t = b_r^t - b^t + \Delta b^{t+1}$.
Call Pat-Gen-Sel algorithm.

4. If $\Delta b^1 > 0$, then decide which products of $d_{k'}^1$ to purchase and which to non-fulfill.

The LS-CSP algorithm calls a pattern generation selection (Pat-Gen-Sel) algorithm to select cutting pattern based on one or more of the following pattern selection criteria:

1. Pattern with minimum loss.
2. Pattern that produces highest number of the piece with the highest demand.
3. Pattern that produces lowest number of the piece with the highest demand.
4. Pattern that produces highest number of the piece with the longest required length.
5. Pattern that produces lowest number of the piece with the longest required length.
6. Select pattern randomly.

The number of pattern selection criteria that will be used in the algorithm will be fixed by setting the maximum number of pattern selection criteria index in Step 2 of the LS-CSP algorithm. To enhance the practicality of the algorithm, whenever a selected pattern generates surplus piece(s), the extra pieces will not be physically cut and the remaining stocks are considered as leftovers that can be reused to satisfy the remaining demand of pieces of current period. Thus, cutting patterns for these leftovers will be determined and added to the available patterns of the standard stock(s) for consideration in subsequent cuts. The Pat-Gen-Sel algorithm proceeds as follows:

1. Generate patterns a_{ij} using one of the following approaches:
   a- Total enumeration method (Suliman (2001)).
   b- Partial enumeration (Gilmore and Gomory (1961)).
2. Increment the pattern selection criteria index.
3. Use the current pattern selection criterion to select a pattern (j).
4. Establish the number of stocks (aluminum sections) cut according to pattern j in period t $\left(s_j^t\right)$, and establish the leftover stocks that can be reused.
5. Adjust lot sizes (x_k^t) and the remaining demand (d_k^t) of the current period.
6. Repeat steps (3) to (5) until satisfying all d_k^t or exhausting of the available capacity of the current period t.
7. Establish the total trim loss for the current period.
8. Repeat steps (2) to (7) until the maximum number of pattern selection criteria is used.
9. Select the cutting pattern solution with the minimum total trim loss.

4. Example Problem

A small instance problem is considered which is used to test the performance of the LS-CSP algorithm with that of the NLIP model developed in Section 2. The algorithm is coded using MATLAB, whereas the optimal solution of the NLIP model is provided by GAMS. Real data from a fabrication shop is used for the

example data. Table 1 shows the data of the problem, it includes 3 final products over 3 scheduling periods, and each product contains a number of units ranging from 0 to 3 of each of the three different piece lengths. The demand of final products and details of cost data are included in the table. Two capacity versions over the three planning periods are used for the example problem: relaxed capacities (500, 500, 500); and tight capacities (300, 300, 250).

Table 2 shows the results for both the algorithm and NLIP model under both capacity versions. The table shows for each period: number of unit's produced of final products; number of unsatisfied units of final products; number of cutting patterns used, the percentages of total trim and surplus, as well as the total costs. With relaxed capacities the algorithm satisfies the demands of final products over all periods, where the NLIP model shows total unsatisfied demand of 20 units from 68 required units in period 2. This unsatisfied demand occurs as a result of the fact that the non-fulfillment costs are low while the purchasing costs are high as indicated in Table 1. The percentages of trim loss and total cost generated by the algorithm are higher than those of NLIP model by about 4.5% and 3.3% respectively.

For tight capacities both the algorithm and the NLIP model fail to satisfy demands by 87 and 63 units, respectively, for a total demand of 207 units over the 3 periods. The trim loss percentage of the algorithm is less than that of the NLP model by 1.5% while the total cost of the algorithm is more by 10.7%. This high difference in the total cost is due to the fact that the algorithm has resulted into surplus of pieces with a length of 20% of the total required length.

On the comparison of the solutions obtained for the example problem by the algorithm and NLIP model, it can be stated that the results from both approaches are comparable.

5. Performance of the Algorithm

Performance of the algorithm is manifested through A set of numerical results. The number of different piece lengths (P) is varied from 3 to 10 (i. e. P = 3, 4, ….., 10) to produce 8 subsets. Each subset includes 5 instance replicates of the combined lot sizing and cutting stock problem, thus resulting into 40 instances for the first set. Each instance includes 5 final products that are needed in each of 6 scheduling periods. In the second set the number of the final products (M) is varied from 5 to 15 (M = 5, 7, 9, 11, 15) to produce 5 subsets. Each subset contains 5 instances, thus resulting into 25 instances for the second set. Each final product is composed of 5 different piece lengths for all the 6 scheduling periods. In the third set the number of scheduling periods (T) is varied from 6 to 24 (T = 6, 12, 18, 24) to produce 4 subsets. Each subset contains 5 instances resulting into 20 instances for the third set. Each instance includes 5 final products, and each product is composed of 5 different piece lengths.

The piece lengths and the demand levels of final products are determined randomly from a range of 10 to 100. This range of lengths and demand is a typical range in the fabrication shops of aluminum sections (frames of windows, kitchens, and furniture). The number of units of each piece length in a final product is established randomly from a range of 0 to 9. The available capacity in each of the scheduling periods is kept constant and equal to 50000 units for all instances and sets. The cost data of the example problem given in Table 1 is used for all instances.

The average results of the three sets of instances are shown in Tables 3-5 for various number of different piece lengths, final products; and scheduling periods, respectively. In each of these tables columns 2, 3, and 4 show the problem size, whereas the remaining columns show the average performance of the algorithm in terms of trim loss, surplus resulting from production of more pieces than required, number of patterns used in the solution, number of setups, and computation time.

In Table 3, the highest trim loss is 2.23% and the highest surplus is 12.45%. The table shows two subsets with trim loss percentage exceeding 1% and the remaining subsets with lesser values. With regard to the surplus: two subsets have values higher than 10%, six subsets have values lower than 5% three of which with values of less than 1%. As expected the available number of patterns increases remarkably with the increase of number of different pieces, however, the highest number of patterns used per scheduling period is 12. It is noticed that the trim loss for problem subsets with relatively high number of different pieces is small compared with subsets of low number of pieces. This is mainly related to the large number of

available patterns for instances with high number of different pieces. As the number of pieces increases, the number of available patterns increases leading to an increase in the algorithm computation time (CPU) as exhibited in Table 3 where the average increases from 2.382 seconds for 3 different piece lengths to 84 seconds for 10 different piece lengths.

In Table 4, the percentages of trim loss and surplus are less than 1% and 3.8%, respectively, for all levels of final products tested. It is noticed that although the number of different pieces remains constant in Table 4, the number of available patterns decreases with the increase of number of final products. This is mainly due to the reduction in number of the different lengths of leftovers, i.e. number of various stock lengths (standard plus leftovers) decreases with the increase of number of final products. The sensitivity of the algorithm computation time to changes in the number of final products is not significant as that with changes in number of different pieces.

Changes in the number of scheduling periods, as manifested in Table 5, show average values for trim loss of less than 1% and for surplus of less than 8%. The effect of number of scheduling periods on number of available patterns is similar to that of number of final products shown in Table 4. The algorithm computation time shows considerable increase with the increase of number of scheduling periods. It should be mentioned that the maximum number of patterns used in all instances does not exceed 12 patterns per period, and the average is 8 whereas the available number of patterns is 74 for 3 pieces and increases to 77545 for 10 pieces.

The performance of the algorithm with respect to trim loss is quite favorable when compared with that exercised in the aluminum fabrication industry which ranges between 3 to 8%. Results of the computation time are quite reasonable for an off-line planning and operational activities.

5. Conclusion

The combined lot sizing and cutting stock problem is formulated as a non-linear integer model that is difficult to solve for most typical practical problems. To overcome this difficulty, an algorithmic solution approach is presented. The algorithm proceeds backward from the last planning period to the first period establishing the products to be produced, their quantities and the cutting patterns to be used for each period under demand and capacity constraints. An example problem is solved by both the algorithm and the non-linear model and the results are presented. Furthermore, using 85 instances, the performance of the algorithm is assessed in terms of trim loss and computation time for a range of different piece lengths; number of final products; and number of scheduling periods. Good results are produced when compared with the practiced performance.

References

1. Heassler, R.W., 1971. A heuristic programming solution to a nonlinear cutting stock problem. Management Science 7(12), 793-802.
2. de Carvalho, J.M.V., Rodriguez, A.J.G., 1995. An LP-based approach to a two stage cutting stock problem. European journal of Operational Research 84, 580-589.
3. Krichagina, E.V., Rudio,R., Taksar, M.I., Wein, L.M., 1998. A dynamic stochastic stock cutting problem. Operations Research 46(5), 690-701.
4. Hendry, L.C., Fok, K.K., Shek, K.W., 1996. A cutting-stock and scheduling problem in the copper industry. Journal of the Operational Research Society 47, 38-47.
5. Nonas, S.L, Thorstenson, A., 2000, A combined cutting-stock and lot-sizing problem. European Journal of Operational Research 120(2), 327-342.
6. Murty, K.G., 1986. Solving the fixed charge problem by ranking the extreme points. Operations Research 16, 268-279.
7. Arbib, C., Marinelli, F, 2005. Integrating process optimization and inventory planning in cutting-stock with skiving option: An optimization model and its application. European Journal of Operational Research 163(3), 617-730.
8. Gramani, M.C.N, Franca, P.M., 2006. The combined cutting stock and lot-sizing problem in industrial processes. European Journal of Operational Research 174(1), 509-521.

9. Gilmore, PC, Gomory, R.E, 1961. A linear programming approach to the cutting stock problem. Operations Research 9, 849-859.
10. Nonas, S.L., Thorstenson, A., 2008. Solving a combined cutting-stock and lot-sizing problem with column generating procedure. Computers and Operations Research 35, 3371-3392.
11. Gramani, M.C.N., Franca, P.M., Arenales, M.N., 2009. A lagrangian relaxation approach to a coupled lot-sizing and cutting stock problem. International Journal Production Economics 119, 219-227.
12. Erjavee, J., Gradisar, M., Trkman, P., 2012. Assessment of stock size to minimize cutting stock production costs. International Journal of Production Economics, 135(1), 170-176.

Table 1 Test Example Data

Number of time periods, T:	3	Number of available stocks per period t, ST(t):	1000	Assembly cost per final product k, cf(k):	[7.5 9.0 10.5]
Number of final products, K:	3	Length of stock, L:	600	Inventory cost per unit stock, hm:	1
Number of different part lengths, I:	3	Capacity per period t, b(t):	[500 500 500]	Inventory cost per unit part, hp:	0.2
Demand of final product k in period t, d(t,k) :	[20 24 23; 18 30 20; 22 25 25]	Setup cost for pattern changing, csj:	100	Inventory cost per unit final product, hf:	0.5
Length of part i, l(i):	[20 30 55]	Cost per unit material length, cm:	0.1	Purchasing cost per unit part, cpur:	15
Number of units of part i per unit of final product k, r(i,k):	[2 2 2; 0 1 2; 3 3 3]	Processing cost per part, cp(i):	0.6 for all parts	Non-fulfillment cost per unit final product k, cn(k):	[2.5 2.5 3.0]

Table 2 Test Example Results

	LSCSP Algorithm	NLP Model	LSCSP Algorithm	NLP Model
Capacity available in period t, b(t):	[500 500 500]	[500 500 500]	[300 300 250]	[300 300 250]
Units produced of final product k in period t, x(t,k):	[20 24 23; 18 30 20; 22 25 25]	[20 24 23; 13 21 14; 22 25 25]	[0 23 23; 0 26 20; 0 12 25]	[20 24 23; 11 19 13; 11 12 12]
Unsatisfied units of final product k in period t, d1(t,k):	[0 0 0; 0 0 0; 0 0 0]	[0 0 0; 5 9 6; 0 0 0]	[20 1 0; 18 4 0; 22 13 0]	[0 0 0; 7 19 7; 11 13 13]
Patterns used out of possible patterns (Number):	[j1, j6; j1 j6; j1 j6] (2 out of 6)	[j4, j4; j4 j6; j4 j6] (2 out of 6)	[j1, j3, j4, j6; j1, j2, j4, j6; j1, j2, j4, j6]	[j1, j2, j4, j6; j4, j6; j4, j6]
Total length of trim (%):	**15.199**	**10.730**	**9.772**	**11.289**
Total length of surplus (%)*:	5.234	0.264	19.569	0.035
Total length of shortage (%)				-1.184
Total length of surplus and trim (%):	20.433	10.994	29.340	11.324
Total cost of material used, Cmat:	5480.00	5400.00	3940.00	3481.14
Total cost of setups, Cset:	600.00	592.58	1200.00	765.45
Total cost of processing, Cproc:	2651.40	2480.11	1922.40	1828.77
Total cost of inventories, Cinv:	195.00	118.72	197.40	420.14
Total Cost of purchased parts, Cpur:			6120.00	
Total cost of nonfulfilled orders, Cnful:	0.00	51.43	195.00	163.565
Total cost, Ctot:	**8796.40**	**8642.83**	**7259.80**	**6659.07**

* Surplus is manifested in short stocks, i. e. surplus is not physically cut from the stock.

Table 3 Average performance of the algorithm for different number of part lengths

Prob. Subset	Problem Size			Trim Loss		Surplus		Trim and Surplus		No of Patterns		No of Setups		CPU*
	No of Periods	No of Products	No of Diff. Pieces	Length	Percent	Length	Percent	Length	Percent	Range Used	Available	Total	Average/period	Seconds
1	6	5	3	2346	0.61	13389	3.44	15735	4.05	5 to 7	74	35	6	2.382
2	6	5	4	6569	2.23	36590	12.45	43159	14.68	5 to 8	1790	39	6	3.566
3	6	5	5	2446	0.70	13113	3.73	15558	4.42	8 to 10	8750	56	9	5.614
4	6	5	6	6011	1.60	34464	9.16	40475	10.76	10 to 12	4983	54	9	8.637
5	6	5	7	8429	0.70	7129	1.38	15558	4.42	8 to 10	16233	58	10	11.679
6	6	5	8	787	0.13	3954	0.67	4740	0.80	8 to9	72115	50	8	15.647
7	6	5	9	672	0.11	3359	0.78	4030	0.67	7 to 8	69691	48	8	45.433
8	6	5	10	842	0.14	4209	0.70	5050	0.84	8 to 9	77545	53	9	84.179
Average				**3513**	**0.78**	**14526**	**4.04**	**18038**	**5.08**		**31398**	**49**	**8**	**22.142**

* Computation Time

Table 4 Average performance of the algorithm for different number of final products

Prob. Subset	Problem Size			Trim Loss		Surplus		Trim and Surplus		No of Patterns		No of Setups		CPU
	No of Periods	No of Products	No of Diff. Pieces	Length	Percent	Length	Percent	Length	Percent	Range used	Available	Total	Average/period	Seconds
1	6	5	5	2446	0.70	13113	3.73	15558	4.42	8 to 10	8750	56	9	10.788
2	6	7	5	1176	0.22	6331	1.21	7507	1.43	2 to10	1562	37	6	17.082
3	6	9	5	3428	0.59	17190	2.96	20618	3.55	2 to 12	2959	41	7	17.932
4	6	11	5	5881	0.49	35647	2.99	41528	3.49	5 to 11	310	48	8	17.722
5	6	15	5	5013	0.26	25487	1.32	30500	1.58	7 to 12	943	59	10	18.763
Average				**3589**	**0.45**	**19554**	**2.44**	**23142**	**2.89**		**2905**	**48**	**8**	**16.457**

Table 5 Average performance of the algorithm for different number of scheduling periods

Prob. Subset	Problem Size			Trim Loss		Surplus		Trim and Surplus		No of Patterns		No of Setups		CPU
	No of Periods	No of Products	No of Diff. Pieces	Length	Percent	Length	Percent	Length	Percent	Used	Available	Total	Average/period	Seconds
1	6	5	5	2446	0.70	13113	3.73	15558	4.42	8 to 10	8750	56	9	10.788
2	12	5	5	1397	0.28	16466	3.28	17863	3.55	2 to 11	858	87	7	55.474
3	18	5	5	1226	0.25	22992	4.68	24218	4.93	3 to 12	1021	123	7	101.736
4	24	5	5	1485	0.31	38845	8.13	40330	8.45	3 to 11	592	160	7	151.867
Average				**1638**	**0.38**	**22854**	**4.96**	**24492**	**5.34**		**2805**	**107**	**8**	**79.966**