



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Difference Process for Solving Non-guillotine Trim Loss Problem

Mahmoud Zennaki and Kaddour Sadouni
Department of Computer Science, Faculty of Science,
University of Science and Technology of Oran (USTO) BP 1505 El-M'naouer, Oran, Algeria

Abstract: Cutting Stock Problems (which are NP-hard), have numerous applications in industry (iron, steel, glass, wood, paper industry) and in studies of computer operations (multi-programmed computer systems, multiprocessor systems). The crucial step in this problem is generating cutting patterns, especially when we have a non-guillotine cutting. This kind of problem also called trim loss problem have several methods: Bottom Left, Best Fit, Difference Process. We propose an algorithm of generating patterns using Difference process method which guarantee improved non-guillotine cutting patterns.

Key words: Cutting stock problem, guillotine and non-guillotine cutting, trim loss problem, difference process, best fit, bottom left

INTRODUCTION

The Cutting Stock Problem (CSP) arises in many mass production industries, when small items (forms) are being cut out from large objects (gross proceeds). The first formulation of the CSP was produced by Kantorovich (1939), leading to the following Integer Linear Programming problem as detailed by Benhamamouch (1979):

$$\left[\begin{array}{l} \text{Min } z = \sum c_j x_j \quad (j=1, \dots, n) \\ \sum a_{ij} x_j \geq q_i \quad j=1, \dots, n \text{ et } i=1, \dots, m \\ x_j \geq 0 \text{ et } x_j \in \text{IN} \quad \forall j=1, \dots, n \end{array} \right.$$

When:

- $M = (L, H)$ dimensions of large objects to be cut.
- $d_i = (l_i, h_i)$ ($i = 1, \dots, m$) dimensions of small items to satisfy in quantity q_i .
- The matrix $A = (a_{ij})$ is built such as each column is a generated cutting pattern.
- The solution is represented by using x_j times of the j th cutting pattern ($j = 1, \dots, n$ where, n is the number of generated cutting patterns). c_j is its correspondent loss.

The CSP have two parts. The first one is the assortment problem addressing the issue of choosing proper dimensions for the large objects. The second one is the trim loss problem addressing the issue of how to

cut out the small items from the given large objects in such a way that wastage material will be minimized (Karelahti, 2002).

MATERIALS AND METHODS

Related works: The Cutting Stock Problem has two difficulties:

- Automatic generation of high quality cutting patterns.
- The number of cutting patterns which increases exponentially with the number of entities to place.

The use of Meta-heuristics (Tabu search, Simulated annealing, Genetic algorithms ...) usually generates faster an acceptable solution. But the solution quality depends essentially of the quality of generated cutting patterns. The case of guillotine cutting has been widely studied by Benhamamouch (1979), Zissimopoulos (1984), Zennaki (1998) and Black (2005). However current industries use more sophisticated cutting tools which can perform a non-guillotine cutting.

Guillotine and non-guillotine cuttings are illustrated in Fig. 1. Hereafter, Gray areas represent trim loss.

It's clear that the amount of generated guillotine cutting patterns is lower than those generated by non-guillotine cutting. In this study we are interested by generating higher quality of non-guillotine cutting patterns by using difference process approach.

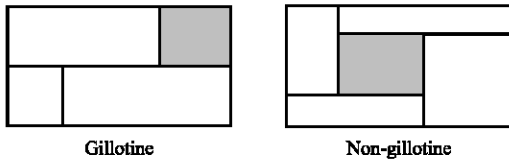


Fig. 1: Guillotine and non-guillotine cutting

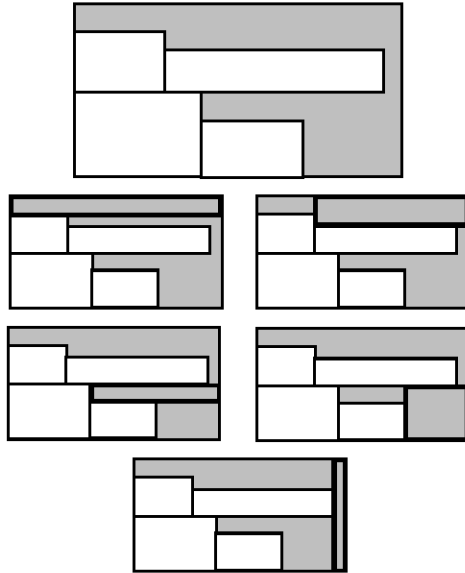


Fig. 2: Difference process approach

Trim loss problem

Bottom left: We can assimilate this algorithm to the well known Tetris game (Leung *et al.*, 1997). First, the form is placed in the right superior coin of the gross proceeds. Next the form is moved as possible to the bottom, then as possible to the left. These moves are repeated until obtaining a stationary state (i.e., the form can't move). This algorithm has been improved by Liu and Teng (1999) by including rotation of forms.

Best fit: The best fit algorithm is inspired from the techniques used in operating system (Leung *et al.*, 1997). A list called loss is updated in order to contain the losses in the actual pattern cutting. The best fit (i.e., the smallest loss) in this list is chosen when a candidate form is to be placed.

Difference process: The main drawback of the precedent approaches is that they can only deliver guillotine cutting patterns. The D.P. algorithm is different in the way that we consider all the available space in the gross proceeds (Liu and Teng, 1999) and can consequently deliver non-guillotine cutting pattern.

The loss in the cutting pattern (Fig. 2) is viewed as a set of 5 losses (the largest rectangular spaces). After

placing a form, the spaces must be re-calculated. Thus, the obtained cutting patterns are better than those obtained by bottom-left or best-fit algorithms and are furthermore non-guillotine.

DIFFERENCE PROCESS ALGORITHM

Representation of cutting pattern: We represent a cutting pattern on a gross proceeds (L, H) in a orthogonal reference (O, X, Y). Thus each form (l, h) is represented by its coordinates (x, y) as shown in Fig. 3.

Localisation: The first part of our algorithm is the localisation of all potential points in which we can place the next form. First, we consider the horizontal (BH = {bh_i}) and the vertical (BV = {bv_i}) bands of the current cutting pattern, then we determine the potential points in each band, respectively named EMPL_{BH} and EMPL_{BV} (Fig. 4).

Initially the single potential point is (0,0), i.e.,

$$BH = \{0\} \quad BV = \{0\} \quad EMPL_{BH} = \{(0,0)\} \quad EMPL_{BV} = \{(0,0)\}$$

The cutting pattern is represented by the array ACT:

Coordinates	Forms
(x, y)	(l, h)

Horizontal and vertical bands are respectively:

$$BH = \{0\} \cup \{ (y + h_i) / [(x, y), (l_i, h_i)] \in ACT \}$$

$$BV = \{0\} \cup \{ (x + l_i) / [(x, y), (l_i, h_i)] \in ACT \}$$

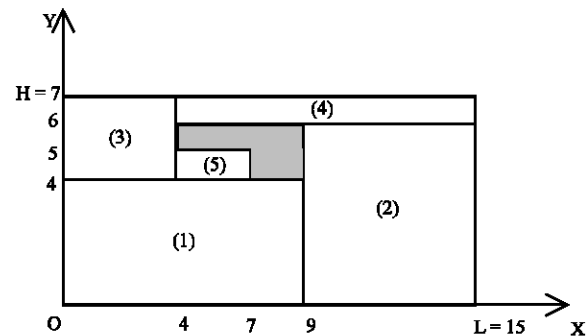


Fig. 3: Representation of cutting patterns

Coordinates (x, y)	Forms (l, h)
(0, 0)	Form 1 (9, 4)
(9, 0)	Form 2 (6, 6)
(0, 4)	Form 3 (4, 3)
(4, 6)	Form 4 (11, 1)
(4, 4)	Form 5 (3, 1)

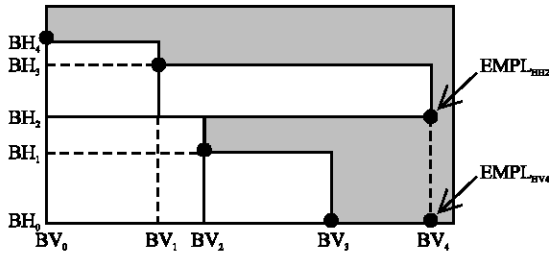


Fig. 4: Localisation of potential points

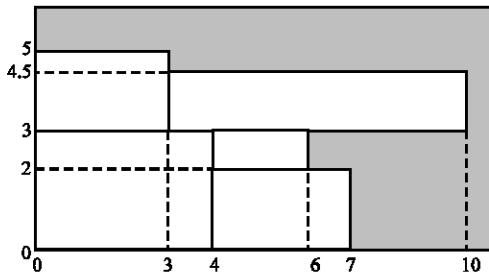


Fig. 5: Example of D.P cutting pattern. Represented by ACT

(x, y)	(l _i , h _i)
(0, 0)	(4, 3)
(0, 3)	(3, 2)
(4, 0)	(3, 2)
(3, 3)	(7, 1.5)
(4, 2)	(2, 1)

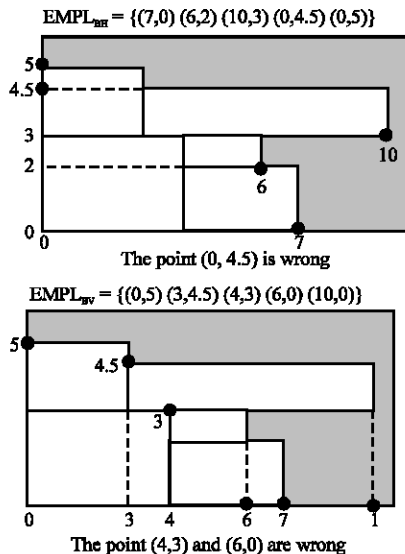


Fig. 6: Illustration of Step 1

The element $bh_i = H$ (respectively $bv_i = L$) is eliminated from BH (respectively BV).

The determination of potential points $EMPL_{BH}$ and $EMPL_{BV}$ is realised in two steps.

Step 1: Determination of all potential points (possible and impossible).

For each horizontal band bh_i :

- Search element of ACT such as $y = bh_i$ with greatest value of x .
- If such element exist (l_i, h_i) then add the point $(x+l_i, bh_i)$ to $EMPL_{BH}$.
- Else add the point $(0, bh_i)$.

For each vertical band bv_i :

- Search element of ACT such as $x = bv_i$ with greatest value of y .
- If such element exist (l_i, h_i) then add the point $(bv_i, y + h_i)$ to $EMPL_{BV}$.
- Else add the point $(bv_i, 0)$.

Example: Consider the cutting pattern $(L, H) = (11, 6)$ as shown in (Fig. 5).

The horizontal and vertical bands are:

$$BH = \{0, 3, 5, 2, 4.5, 3\} = \{0, 2, 3, 4.5, 5\}$$

$$BV = \{0, 4, 3, 7, 10, 6\} = \{0, 3, 4, 6, 7, 10\}$$

After execution of step 1, the potential points obtained are shown in Fig. 6.

Step 2: Pre-test of potential points (Test before placement of the form). This step is a correction of Step 1 which can give some wrong points.

A point (a, b) is wrong if: $\exists [(x,y), (l_i, h_i)] \in ACT$ such as: $x \leq a < x+l_i$ and $y \leq b < y+h_i$

The correction of wrong points is detailed as follows:

- $EMPL_{BH}(\alpha, bh_i)$
 $t = \alpha$
 While (t, bh_i) wrong and $t < L$ do
 - Let $[(x,y), (l_i, h_i)]$ the element of ACT which contains the wrong point:
 $t = x+l_i$
 EndWhile
 $\alpha = t$ (* Correction *)
- $EMPL_{BV}(bv_i, \alpha)$
 $t = \alpha$
 While (bv_i, t) wrong and $t < H$ do
 - Let $[(x,y), (l_i, h_i)]$ the element of ACT which contains the wrong point:
 $t = y+h_i$
 EndWhile
 $\alpha = t$ (* Correction *)

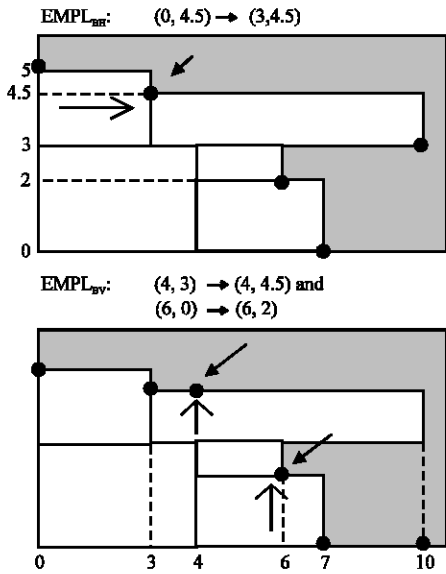


Fig. 7: Illustration of Step 2

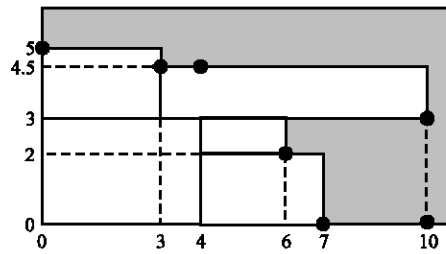


Fig. 8: Localisation of potential points

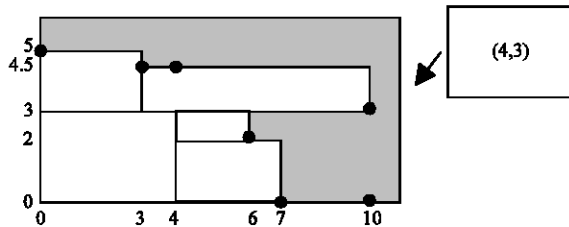


Fig. 9: Placement of form (4,3)

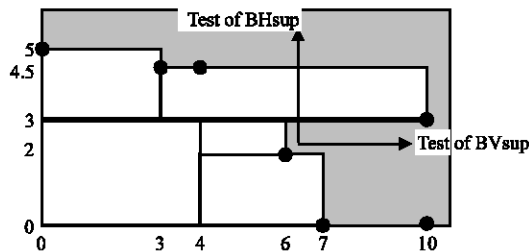


Fig. 10: Post-test

The correction of wrong points in the previous example give results as shown in Fig. 7.

At the end of this step we set $EMPL = EMPL_{BH} \cup EMPL_{BV}$

Thus $EMPL = \{(7,0)(6,2)(10,3)(3,4.5)(0,5)(4,4.5)(10,0)\}$

Remark: The points (10, 3) and (4, 4.5) are useful for the second part of our algorithm but ignored in the final (Fig. 8)

Placement of the form: After localisation of potential points (EMPL), we continue the test by trying the placement of the form in each potential point. For that, two tests are realised:

Overflow test: This test eliminates potential points which cause overflowing. Let take the case of disposing the form (4,3) in the following cutting pattern (Fig. 9).

The set of potential points is $EMPL = \{(0, 5)(7, 0)(3, 4.5)(6, 2)(4, 4.5)(10, 0)(10, 3)\}$

While trying to place the form in each point, we obtain:

$(0, 5) + (4, 3) = (4, 8)$	Overflow
$(7, 0) + (4, 3) = (11, 3)$	OK
$(3, 4.5) + (4, 3) = (7, 7.5)$	Overflow
$(6, 2) + (4, 3) = (10, 5)$	OK
$(4, 4.5) + (4, 3) = (8, 7.5)$	Overflow
$(10, 0) + (4, 3) = (14, 3)$	Overflow
$(10, 3) + (4, 3) = (14, 6)$	Overflow

Thus the set EMPL become $\{(7, 0)(6, 2)\}$.

Post-test: We do this test in order to eliminate wrong points which appear after placement of the form. In the precedent example, it's impossible to place the form (4,3) at the point (6,2) even if it doesn't cause overflowing. To move away these kinds of points, the Post-Test consist to look through the vertical and horizontal bands (Fig. 10).

The potential point (6,2) is eliminated by the Post-Test because of horizontal band BH3 ($EMPL_{BH3} = (10,3)$).

The post-test is realised as follows:

Let (X, Y) a potential point to be tested,

Test BVsup

Search $bv_i \in BV$ such as $bv_i > X$

If no bv_i exist then (X,Y) is correct

Else

Let (bv_i, α) the correspondent element in $EMPL_{BV}$

If $\alpha \leq Y$ Then Test BHsup

Else If $X + l_i \leq bv_i$ then Test Bhsup

Else (X,Y) is wrong

Test BHsup

Search $bh_i \in BH$ such as $bh_i > Y$
 If no bh_i exist then (X, Y) is correct
 Else
 Let (α, bh_i) the correspondent element in $EMPL_{BH}$
 If $bh_i \geq Y + h_i$ then Test next BV_{sup}
 Else If $\alpha \leq X$ then Test next BH_{sup}
 Else (X, Y) is wrong

RESULTS AND DISCUSSION

The results confirm the high quality of cutting patterns. Furthermore, no constraint in the cutting process is imposed (Non-guillotine). This is illustrated in the following example:

Example: Let dispose the following forms in a gross proceeds $(L, H) = (15, 7)$

l_i	h_i
9	3
6	6
4	4
11	1

The placement of the 3 first forms lead to the cutting pattern as shown in Fig. 11.

At this stage, our algorithm give 3 potential points: $EMPL = \{(4, 3) (4, 6) (9, 6)\}$
 While disposing the last form $(11, 1)$ we have:

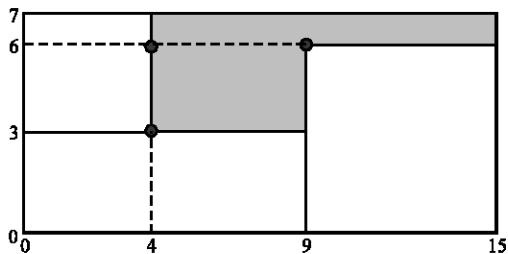


Fig. 11: Current cutting pattern

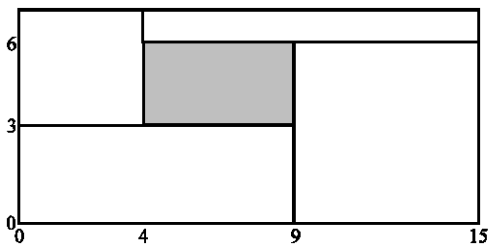


Fig. 12: Non-guillotine cutting pattern

- $(4, 3)$ is wrong because of $BV_{sup} = 9$.
- $(9, 6)$ cause overflowing.
- $(4, 6)$ is correct. This leads to the cutting patterns which cannot be obtain with a guillotine cutting process (Fig. 12).

CONCLUSIONS

The automatic generation of non-guillotine cutting patterns has been our principal goal. The proposed algorithm can give high quality solutions in the case of 2-dimension rectangular forms. We will extend our work to an efficient heuristic which can generate a sufficient number of non-guillotine cutting patterns for solving the CSP problem.

REFERENCES

Benhamamouch, D., 1979. An optimal CSP algorithm in IR^2 . Ph.D Thesis. Pierre and Marie Curie University, Paris, 6.

Black, P.E., 2005. Cutting stock problem. In Dictionary of Algorithms and Data Structures. Available from: <http://www.nist.gov/dads/HTML/cuttingStock.html>

Kantorovich, L.V., 1960. Mathematical methods of organizing and planning production. Manage. Sci., An English translation of a Russian paper published in 1939, 6: 363-422.

Karelahti, J., 2002. Solving the cutting stock problem in the steel industry. M.sc. thesis. Espoo.

Leung, T.W., C.H. Yung and M.D. Troutt, 1997. Applications of genetic search and simulated annealing to the dimensional non-guillotine cutting stock problem. Graduate School of Management, Kent State University.

Liu, H. Teng. 1999. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. Eur. J. Operat. Res., 112: 413-420.

Zennaki, M., 1998. Tabu search and simulated annealing for solving CSP. M.Sc Thesis. U.S.T.O.M.B.

Zissimopoulos, V., 1984. Approximate Algorithm for CSP. Ph.D Thesis. Orsay.