Revista de Matemática: Teoría y Aplicaciones 2013 **20**(1) : 35–48
cimpa – ucr issn: 1409-2433

ONE-SIDE OSCILLATION STRATEGIC APPROACH

ENFOQUE ESTRATÉGICO DE OSCILACIÓN UNILATERAL

Ricardo Beausoleil* Yasser Valcárcel Miró†

Received: 17/Sep/2011; Revised: 3/Oct/2012;
Accepted: 27/Nov/2012

---

*Departamento de Matemática Interdisciplinaria, Instituto de Cibernética, Matemática y Física. Calle E # 551, Vedado, Ciudad de La Habana, CP 10900 Cuba. Fax: +(537) 832 4085. E-Mail: rbeausol@icimaf.cu

†Misma dirección que/*Same address as:* E-Mail: yasserv@icimaf.cu

Abstract

This paper reports an approach developed to find a good quality solution for a generalized assignment problem with application to the *Dimensional Cutting Problem*. The objective at the *Cutting Problem* is to divide different pools of pieces of the same dimension among different items of available material all having two dimensions, the width and length. The solution is found keeping the largest number of pieces in a single material. Some initial numerical experiences found for the algorithm optimal and suboptimal solutions or instances of more than 500 pieces at a very low computational cost.

Keywords: heuristics, greedy algorithm, backtrack algorithm, one-dimensional cutting problem.

Resumen

El presente artículo presenta un enfoque desarrollado para encontrar una solución de alta calidad para un problema de asignación generalizado con aplicaciones al Problema de Cortes Dimensionales. El objetivo en el *Problema de Cortes* es dividir diferentes juegos de piezas de las mismas dimensiones dentro de varios artículos de material disponible, teniendo ambos dos dimensiones: ancho y largo. La solución es tomada manteniendo el mayor número de piezas en un sólo material. Algunas experiencias numéricas encontraron soluciones óptimas y cuasi-óptimas para el algoritmo en no más de 500 piezas con un costo computacional bajo.

Palabras clave: Heurísticas, algoritmo goloso, algoritmo backtrack, problema unidimensional de cortes.

Mathematics Subject Classification: 97M40, 90C59, 68T20.

1 Introduction

The difficulties of cutting yield often on the efficient use of the materials. One way to overcome this objective is to allocate the pieces in the materials having to be cut in such a way that either the rest or each length of material is minimized. A heuristic to the *One Dimensional Cutting Problem* was described in [1]. In this paper we report a *one-side strategic oscillation approach* [4] heuristic technique for the linear *Two Dimension Cutting Problem* (length and width) for the materials and pieces. This approach operates by alternating constructive and destructive phases where each solution generated by a constructive phase is dismantled to a variable

degree by a destructive phase, after which a new constructive phase builds the new solution. The complexity of any procedure for solving this class of problems is expected to be exponential in the number of pieces and the materials involved.

The present paper is divided into 7 sections, section 2 introduces the One Dimensional Cutting Problem as section 3 presents a model describing the DCP, then 2 heuristic approaches are schemed in section 4 for the latter presentation of a formal procedure solving the DCP. Section 6 then presents some experimental results of a particular instance of the *Cutting Problem* and section 7 summarizes the results of the present algorithm solving the One Dimensional Cutting Problem by means of a one-side oscillation strategic approach.

2 Problem definition

A formal introduction of the two dimension cutting stock problem will be presented as an *Assignment Problem* at the present section.

We are given a set of indices of materials $R = \{1, 2, \ldots, m\}$, let us consider one type of material with properties of uniformity, such as regular geometric shape and two dimensions, a set of indices of type of a piece or $T = \{1, 2, \ldots, q\}$, a list of q demands of pieces of different types $D = \{d_1, d_2, \ldots, d_q\}$.

Associated to each type of material we have s pools of pieces, each one of them composed of pieces of the same dimension; s being the number of c_{wt} equal to the material consumed by each piece associated to a material of width w and type t. Let us denote by w_r the width of material r, by l_r it's length, by c_{rt} the length of material r consumed by each type of piece t. Let us as well define as p_{it} the number of pieces of type t with the width i. Consider that:

$$\sum_{i=1}^{s} p_{it} \leq d_t \quad \forall t \in T$$

and $p_{it} > 0, \forall t \in T, i = 1, \ldots, s$. We wishato assign the pieces to each item of available material, maximizing the residual material, but minimizing at the same time the items used, for each item of width w_p we could only allocate pieces of the same width of the material selected.

3 Mathematical model

Let then be the following *Mixed-Integer Linear Program*:

- $R = \{1, 2, , \ldots, m\}$ set of indices of items of materials,
- $G = \{1, 2, , \ldots, m\}$ set of indices of items of pieces,
- c_{rg} material r consumed by the piece g.

Defining

$$x_{rg} = \begin{cases} 1, & \text{if the piece } g \text{ is assigned to the material } r \\ 0, & \text{otherwise.} \end{cases}$$

Being:

$$\begin{aligned} &\min \quad \textstyle\sum_R a_r y_r && && (1)\\ &s.t: \\ &\qquad \textstyle\sum_G c_{rg} x_{rg} + y_r && = l_r \quad \forall r \in R \\ &\qquad \textstyle\sum_R x_{rg} = 1 && \forall g \in G && (2)\\ &\qquad x_{rg} \in \{0, 1\} && \forall r \in R, \forall g \in G \\ &\qquad y_r \geq 0 && \forall r \in R \end{aligned}$$

The number of variables increases exponentially with the number of pieces and materials involved. Two heuristic algorithms were developed to solve this problem, one, referred to as dual, is used to obtain a lower bound while a primal one finds an upper bound aiming to be as near to the optimal value as possible. We formulate the dual linear program [2], associated with the LP relaxation to the *Mixed-Integer Program* and solve this heuristically.

Dual Linear Program associated with the LP Relaxation.

$$\begin{aligned} &\max \quad \textstyle\sum_R l_r u_r + \sum_G y_r && && (3)\\ &s.t: \\ &\qquad l_r u_r \leq w_r && \forall r \in R && (4)\\ &\qquad c_{rg} u_r + y_g \leq 0 && \forall r \in R, \forall g \in G && (5) \end{aligned}$$

with $u_r y_r$ unrestricted in sign.

4 Heuristic algorithms

4.1 Dual procedure

In order to obtain a feasible solution we make a partition in G, $D_1 = \{1, \ldots, d_1\}$, $D_2 = \{d_1 + 1, \ldots, d_2\}, \ldots, D_q = \{d_{q-1} + 1, \ldots, d_q\}$, in such a way that:

$$\begin{aligned} D_i \bigcap D_j &= \emptyset \\ \bigcup_{i=1}^{q} D_i &= G. \end{aligned}$$

Then we can write 3 in the following form:

$$c_{rt} u_r + y_t \leq 0, \quad \forall r \in R, \quad t = [1, q] \tag{6}$$

where for each D_t, $c_{rt} = c_{rg}$, for every $g \in D_t$ whenever $r \in R$ is fixed. Then for each $g \in D_t$, c_{rg} is a constant. From 4 we have that $u_r \leq w_r/l_r$, there exists r_0 such that $w_{r_0}/l_{r_0} \leq w_r/l_r$, $\forall r \neq r_0$, then, if we take $u_r = w_{r_0}/l_{r_o} = a_{r_0}$, for all $r \in R$ the constraint 4 holds. Then from 6 we obtain the following inequalities:

$$y_l \leq \min\left\{-c_{r_1 a_{r_0}}\right\}, \quad y_2 \leq \min\left\{-c_{r_2 a_{r_0}}\right\}, \ldots, \quad y_q \leq \min\left\{-c_{r_q a_{r_0}}\right\}$$

Now, if we take $y_t = \min\left\{-c_{r_1 a_{r_t}}\right\}$, $\forall t = [1, q], r \in R$, we obtain a feasible solution. The objective can be written as follows:

$$a_{r_0}\left(\sum_{R} l_r - \sum_{t=1,\ldots,q} d_t c_{r't}\right),$$

where $c_{r't} = \max\{c_{rt}\}, \forall r \in R$ and t fixed, then we have the following lower bound:

$$\mathbf{LB} = \max\left\{0, a_{r_0}\left(\sum_{R} l_r - \sum_{t=1,\ldots,q} d_t c_{r't}\right)\right\}.$$

4.2 Primal procedure

The basic idea of the algorithm is:

1. First to allocate as many pieces as possible in the material of dimension r that has been associated to the maximum value of the coefficient a_r, so as to satisfy: $a_{r_1} \leq a_{r_2} \leq \ldots a_{r_m}$, which is called the **IPWL** or *"Importance per width and length"* rule. Apply it iteratively until no capacity is left to allocate pieces of the selected items. The application of this rule generates a sequence of items, denoted as forward sequence. If more pieces need yet to be assigned, go to (2).

2. Create a back sequence of items of materials in the following form. First, go back to the previous item at the seed point in the current back sequence(at the beginning, the seed point in the back sequence is the last item in the forward sequence), then set a new seed point; at this moment we have a sub-sequence of the forward sequence, forbid the start of an assignment from the pool of pieces that began this sub-sequence and re-install the pieces allocated in your associated pools and then apply the first rule, until we do not have any piece to allocate or all pools that have pieces to assign are forbidden to begin the assignment. If all pools of pieces associated to the selected material were forbidden and the seed point of the forward sequence was not reached, go to (2), otherwise go to (3).

3. If the utility is greater than the lower bound and no pieces were yet to be assigned, then forbid the pool beginning the last iteration and go to (2), otherwise stop. Here we're supposing that we have to test $s \times q$ possibilities, one for each iteration, where s is the width and q the number of pieces.

4. Create a back sequence of items in the following manner: Return to the previous item at seed point on the back sequence, since at the beginning a seed point is the last item on the forward sequence. Then set a new seed point, at this moment we have a subsequence hence re-install the pieces allocated on the associated pools. If at least one pool having pieces to assign is not forbidden to begin the assignment apply (2) if all pools of pieces associated to the selected material were forbidden and the seed point of the forward sequence were not reached go to (4), else go to (3).

4.3 Steps of the method

Step 1 (Initialization)

- Create feasible items of materials.
- Choose the material r_0 that have coefficient $a_{r_0} = \max\{a_r\}$ in such a way that at least one pool of pieces associated to the selected material had not began any previous iteration.
- Choose the pool of pieces in such a way that this pool had not began any iteration in the past and the pieces of this pool could be assigned to the selected material r_0. Initialization of the set S.
- Forbid the selected pool of pieces to begin the following iterations. If all pools of pieces associated to wide w_0 corresponding to selected material r_0 began one iteration in the past, then forbid all items of materials that have been selected of wide w_0 to begin the following iterations.
- If rule $(R1)$ is violated, then create feasible set of materials.

Step 2 (Assignment)

(A) Choose a feasible material r_0 in **IPWL** order. Choose the biggest feasible pool of pieces in such a way that your pieces could be allocated in the selected material. Load the material with the greatest possible amount of pieces of the selected pool. If rule $(R1)$, is violated then create a new feasible set of materials.

(B) (Update) Let us call **UBnow** and **UBnext** the current and next upper bounds respectively. If **Ubnow**<**Ubnext** then reset **Ubnow**=**Ubnext**. If **Ubnow**=**LB** or(variants were always executed) then output the solution and stop, Otherwise return to (A).

(C) (*Destructive Phase*) Apply the rule $(R3)$. Update the materials that will receive pieces, that is, set $r_0 = rs_p$. Apply the $(R2)$ and return to (A).

Rules:

(R1) If we do not have feasible items of materials and we have pieces to assign, then apply a *Destructive Phase*.

(R2) Re-install the pieces that were assigned in the sequences of rolls selected and forbid the greatest size of the pieces contained in the seed point of this sequence.

(R3) Select the rs_p material previous to the seed point in the back sequence, initially the seed point in the back sequence is the last item of material in the forward sequence.

5 Formal procedure

A rather informal description of the solution method presented in the previous sections and subsections will be stated. Then the *Dual Procedure* will be as follows:

- Choose $a_{r_0} = \min\{w_r/l_r\}, \forall r \in R$.
- Set $y' = \max\{c_{rt}, \forall r \in R\}, \forall t \in [1, q]$.
- evaluate $LB = a_{r_0}(\sum_R l_r - \sum_{t=1}^{q} d_t y')$.
- If $LB > 0$ then set it to zero.

Now we proceed to describe a one-side oscillation procedure for our original problem. In such way new definitions are required:

- $R^f = \left\{r : r \in R, l_r \geq \min\left\{c_{rt} : t \in T'\right\}\right\}$.
- $T' = \{t_p, t_{p+1}, \ldots, t_{p+l}\}$, set of type pieces available for assignation.
- S set of assigned pieces.
- LT list of forbidden types of pieces.
- Gb collection of types of pieces not available at the beginning of a iteration.
- RS sequence of items of materials created on the assignment process.

Primal Procedure(One-side oscillation)
begin
 set $Gb = \emptyset$;
 while $Gb \neq s \times q$ do,
 begin

$S = \emptyset$; $R^f = \emptyset$; $LT = \emptyset$; $T^{'} = T$;
Create R^f
if $R^f = \emptyset$ then output(unfeasible solution); Exit;
% Select the material
Choose $r_0 \in R^f : a_{r_0} = \max\{a_r = w_r/l_r \forall r \in R, w_r \notin Gb_w\}$
%The type of the piece associated to material of dimension r_0 selected
Choose $t_0 : \exists t \in T^{'} - Gb, c_{w_0 t_0} = \max\left\{c_{w_0 t} : t \in T^{'} - Gb \wedge c_{w_0 t_0} \leq l_{r_0}\right\}$
% Where w_0 is the wide of the material r_0
Assign all the possible pieces of size t_0 of the selected material r_0;
If the pieces of type t_0 had always been satisfied then $T^{'} = T^{'} - t_0$
$Gb = Gb + t_0$;
Create R^f;
forward;
if $T^{'} = \emptyset$ then output ('solution,S');
else ('a solution was not attained');
if the pieces of type t_0 had been satisfied then $Gb_w = Gb_w + w_0$;
end;
end;

Forward Procedure
begin
while $R^f \neq \emptyset$ & $T^{'} - LT \neq \emptyset$
begin
Choose $r_0 \in R^f : a_{r_0} = \max\left\{a_r, r \in R^f\right\}$
Set $RS(k) = RS(k) + r_0$; update k;
% Select the type of the piece associated to material r_0 selected
Choose $t_0 : \exists t \in T - LT, c_{w_0 t_0} = \max\{c_{w_0 t} : t \in T - LT \wedge c_{w_0 t_0} \leq l_{r_0}\}$
Where w_0 is the width of the material r_0
% The following while repeats while exists demand and the length of
% the selected material be greater than the consumed by the piece
while demand & length do,
begin
Choose $g_{t_0} \notin s$:
Set $S = S + g_{t_0}$;
Create a new R^f;
while $R^f \neq \emptyset \wedge T^{'} \neq \emptyset$ do,
begin
Update the seed point of the back sequence;

```
            goback;
      end;
    end;
  end;
end;
```

```
Goback Procedure
begin
  R^f = R^f + {rs_p for all p on the back sequence };
  T' = {T' + t_p : t_p for all pieces pertaining
  to the back sequence }
  S = S − {g : g was assigned to at least one rs_p pertaining
  to the back sequence }
  LT = LT − max{t_p : t_p type of first item of pieces assigned
  to the back sequence }
  forward;
end;
```

Example 1 *A simple example clarifying the use of the algorithm would be. Suppose the items to be rolls of clothes of different width and length, hence the type are their sizes, also suppose that we had* 3 *sizes and* 3 *rolls. Let's take the widths equal to* 2, 2 *and* 3 *meters respectively, the lengths of* 5, 4 *and* 3 *meters, the sizes ss, s and m, and let finally the demand be of garments of* 4, 3 *and* 2 *respectively.*

Then we have, $T = \{ss, s, m\}$, $R = \{1, 2, 3\}$, $D = \{4, 3, 2\}$ *and* $G = \{1, 2, \ldots, 9\}$. *Let us then express in a* 3×9 *matrix the material consumed for the garments in each roll of different wide which we will call* C.

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 & 1.5 & 1.5 & 1.5 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1.5 & 1.5 & 1.5 & 2 & 2 \\ 0.7 & 0.7 & 0.7 & 0.7 & 1.2 & 1.2 & 1.2 & 1.8 & 1.8 \end{bmatrix}$$

Then $a_1 = 2/5$, $a_2 = 2/4$ *and* $a_3 = 3/3$ *and the system is as follows:*

$$x_{11} + x_{12} + x_{13} + x_{14} + 1.5x_{15} + 1.5x_{16} + 1.5x_{17} + 2x_{18} + 2x_{19} + y_1 = 5$$

$$x_{21} + x_{22} + x_{23} + x_{24} + 1.5x_{25} + 1.5x_{26} + 1.5x_{27} + 2x_{28} + 2x_{29} + y_2 = 4$$

$$0.7x_{31} + 0.7x_{32} + 0.7x_{33} + 0.7x_{34} + 1.2x_{35} + 1.2x_{36} +$$

Wide	Size					
	1	2	3	4	5	6
1	1	1.5	2	2.5	3	3.2

Table 1: Fabric consumed by a set of molds.

Wide	Size						
	1	2	3	4	5	6	7
	151	130	160	125	100	165	70
	8	9	10	11	12	13	-
1	85	70	95	80	88	123	-

Table 2: Length of the rolls.

$$+1.2x_{37} + 1.8x_{38} + 1.8x_{39} + y_3 = 3.$$

The algorithm would set, $x_{38} = 1$ *and then* $y_3 = 1.2$, $x_{35} = 1$ *and then* $y_3 = 0$. *Selecting another roll we would set* $x_{29} = 1$ *and then* $y_2 = 2$, $x_{26} = 1$ *and then* $y_2 = 0.5$, *and finally selecting the last roll,* $x_{17} = 1$ *then* $y_1 = 3.5$, $x_{11} = x_{12} = x_{13} = 1$ *and at last* $y_1 = 0.5$.

In this moment we do not have place to allocate garments, set of modules for garments, but we still have some garments unassigned, then this solution is not feasible, for this reason we destroy this solution using the backtrack algorithm and construct a new solution.

Select $x_{38} = 1$, $x_{35} = 1$ *and then* $y_3 = 0$. $x_{26} = x_{27} = 1$ *then* $y_2 = 1$, $x_{21} = 1$ *then* $y_2 = 0$, $x_{19} = 1$ *then* $y_1 = 3$, $x_{12} = x_{13} = x_{14} = 1$ *then* $y_1 = 0$, *hence obtaining an optimal solution.*

6 Computational experiment

We now illustrate some computational experiments. We had applied our algorithm to the clothing industry; in that problem a piece would be equivalent to the fabric used by the set of models to obtain each model of demanded garment.

Size	1	2	3	4	5	6
Demand	82	75	150	108	57	156

Table 3: Demand of garments.

Size	Roll												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1					81							
2		1	20			53		1		1		1	
3	75	8	65							1			
4				15		50		1			1	1	10
5												25	32
6					7		21	26	21	29	25	27	13

Table 4: Example of optimal distribution of fabrics.

Size	Roll												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1		1		1	78	1						
2		1	18			55		1			1		
3	75	8	66							1			
4			45			50						1	12
5							3					23	31
6							28	26	21	29	25	27	

Table 5: Example of optimal distribution of fabrics.

Six sizes of garments are demanded. We work with materials, in our case rolls of clothes of one dimension of length, for which we consider the width as equal to 1 at each roll. The fabric consumed for a set of molds for each size, is shown inTable 1, while Table 3 shows the demand of each garments. In this example we have thirteen rolls; the length are shown in Table 2.

Tables 4 and 5 show two optimal distributions of the fabric pieces, set of molds, of each size needed to satisfy the demand of garments on the rolls. In this example both the upper and lower bounds were equal to 0.04, for this reason we can say that this result is optimal.

Wide	Roll						
	1	2	3	4	5	6	7
1	70	45		65	40		
2			52			46	50

Table 6: Dimensions of the seven rolls.

Size	1	2	3	4	5
Demand	45	73	56	26	12

Table 7: Demands of sizes.

Roll	Wide I				Wide II		
Size	1	2	3	4	1	2	3
1	30		14			1	
2		30	34	8	1		
3			14		34		8
4						23	3
5						12	

Table 8: Different allocations of the molds.

Table 6 shows seven rolls and two widths and length of each one, while Table 7 shows the demand and Table 8 two different allocations of the set of molds at each roll.

In this experiment we have two widths, 1 and 2 meters respectively, the coefficients of the material consumed were $c_{11} = 1, c_{12} = 1.5, c_{13} = 2, c_{14} = 2.5, c_{15} = 3, c_{21} = 0.6, c_{22} = 1, c_{23} = 1.5, c_{24} = 2$ and $c_{25} = 2.6$.

The spared clothes in both experiments were: In the first allocation $40m$ in the roll of length $70m$ and widths $1m$ and $20cm$. In the roll of length $46m$ and width $2m$, in the second allocation $40m$ in the roll with length $70m$ and width $1m$ and $20cm$ in the roll of length $50m$ and width $2m$. In this example the upper and lower bounds were 0.5 and 0.01 respectively, while the gap between these gives an indication of the quality of the solution obtained.

7 Conclusion

In this paper we considered a *Two-Dimensional Cutting Problem* that asks to maximize the efficient use of the material. The traditional methods give impractical solutions for this problem. Thus, we have proposed a heuristic approach that uses a *Greedy with Backtracking Technique* [3] and shows that it gives optimal or near optimal solutions in a very low computational time (seconds) in a personal computer for problem instances of more than 500 pieces.

References

[1] Beausoleil, R.P. (1997) "Heuristic algorithm for assignment problem with application in the factory of clothes", *Proceedings of the 13th ISPE/IEE International Conference on CAD/CAM Robotics and Factories of the Future*: 995–1000.

[2] Colin, R.R. (1993) *Modern Heuristic Techniques for Combinatorial Problems*. American Halsted Press, John Wiley and Sons Inc., New York

[3] Golomb, S.W.; Baumert L.D. (1965) "Backtrack programming", *ACM Journal* **12**: 516–526.

[4] Portela, J.M.; Kuzmich, V. (1989) *Modelos Económico-Matemáticos*, Tomo I. Editorial Pueblo y Educación.

[5] Simmons, D.M. (1969) "One-dimensional space allocation: an ordering algorithm", *ORSA Journal* **17**(5): 812–826.