Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA. **Rect@**
Volumen 23(2022). Páginas 1 a 22
DOI: 10.24309/recta.2022.23.1.01

CONTRIBUTION TO SOLVING A ONE-DIMENSIONAL CUTTING STOCK PROBLEM WITH TWO OBJECTIVES BASED ON THE GENERATION OF CUTTING PATTERNS

BOUALEM SLIMI
boualemslimi@yahoo.com
Department of Mathematics, Faculty of Sciences, Blida University, Algeria

MONCEF ABBAS
moncef_abbas@yahoo.com
Faculté des Mathématiques, Université des Sciences et Technologies Houari Boumadian,
Laboratoire d'Aide Multicritère à la Décision et Recherche Opérationnelle,
BP32, Bab-Ezzouar,16111, Alger, Algerie

Recibido (28/12/2021)
Revisado (19/08/2022)
Aceptado (05/09/2022)

RESUMEN: En las versiones clásicas del problema del material de corte, el objetivo es encontrar una solución para cortar un objeto principal en varias partes comúnmente llamadas piezas, para minimizar la pérdida total de recorte de la materia prima. Numerosos estudios han abordado este tipo de problema. Sin embargo, en las aplicaciones del mundo real, generalmente existen restricciones que hacen que la forma del problema sea diferente de la versión clásica y que sea más difícil de resolver. En este artículo se propone una técnica para resolver el problema del stock de corte unidimensional con dos objetivos, donde se busca minimizar al mismo tiempo la pérdida total de recorte de la materia prima y el número de setups a realizar. Esta técnica está constituida por dos etapas cuya primera consiste en generar todos los patrones de corte factibles y la segunda permite construir planos de corte, satisfaciendo las demandas, gracias a un subconjunto de estos patrones. Estos diferentes planes de corte representan todas las soluciones factibles, cada una de las cuales se caracteriza por un número de configuraciones y cantidad total de caídas

Palabras Clave: pérdida de recorte, instalaciones, patrón de corte factible, plan de corte factible y un problema de stock de corte unidimensional con dos objetivos..

ABSTRACT: In the classic versions of the cutting stock problem, the aim is to find a solution to cut a main object into several parts commonly called pieces, in order to minimize the total trim loss of the raw material. Many studies have addressed this type of problem. However, in real-world applications there are usually constraints that make the problem shape different from the classic version and make it more difficult to solve. In this article we propose a technique to solve the one-dimensional cutting stock problem with two-objectives, where one seeks to minimize at the same time the total trim loss of the raw material and numbers of setups to be carried out. This technique is constituted of two stages whose first consists in generating all the feasible cutting patterns and the second allows to build cutting planes, satisfying the demands, thanks to a subset of these patterns. These different cutting plans represent all of the feasible solutions, each of which is characterized by a number of setups and total quantity of falls..

Keywords: trim loss, setups, feasible cutting pattern, feasible cutting plan, cutting stock problem with two-objectives.

1. Introduction

The cutting problem represents a family of problems studied in the domain of Operational Research. This problem appears in a large variety of sectors, such as those of the paper industry, aluminum and other metals, glass… etc. The study of these problems initially focused on the study of standard problems, while realistic solution approaches to cutting problems in practice do not only have to provide cutting plans, but also have to deal with other aspects and have to answer additional questions. One such aspect concerns the setups which may arise in industrial cutting processes whenever a new cutting pattern different from its predecessor is started and the cutting equipment has to be prepared in order to meet the technological requirements of the new pattern. Setups of this kind involve the trim loss of production time capacity and the consumption of resources. (Haessler, 1975) introduced a formulation of a one-dimensional cutting stock problem with setup cost where he proposed a pattern generating heuristic algorithm called the sequential heuristic procedure (SHP), where a cutting plan is constructed sequentially by choosing such patterns that can be applied with high frequency and small trim loss, while Diegel et al, (1996) mentioned values for the cost of trim loss (c_1) and cost of setup (c_2) and they showed that there is a relationship between c_1 and c_2 depending on several factors such as: demand, due dates, labor costs… etc

Therefore, the optimization process is based on the presented model, polarizes on a particular type of solution which depends on c_1 and c_2. In this work, we will study the cutting stock problem with setup by the bi-objective approach, making it possible to obtain a set of efficient solutions which are not generally evaluated by a common scalar function. On the other hand, the one-dimensional cutting problem with setup cost expresses two contradictory objective functions, namely the total trim loss of the lost raw material, as well as the number of setups, which makes the use of the technique multi- objective to solve the problem more precise and more appropriate to reality.

For this purpose, we organize this paper as follows. In the next section, we will discuss some of the important work that has accompanied extensions of the cutting stock problem. Then we introduce the definition and the formulation of the one-dimensional cutting stock problem with cost of setup; also present some basic concepts of the multi-objective approach and also the definition and the formulation of multi-objective cutting stock problem with cost of setup. The fourth section is devoted to the resolution approach. In the fifth section we develop an algorithm to solve this problem. In the sixth section, we illustrate the proposed method with examples, and the last section concludes the paper.

2. Literature Review

Cutting problems represent a fertile field of research both in terms of its applications and in terms of its resolution. However, the importance of the problem is even greater at the level of its extensions. This has prompted several researchers to undertake research on this problem.

Within the framework of single-objective optimization Gilmore and Gomory, (1961, 1963, 1965), between the years 1961 and 1965, used linear programming (LP) methods for solving the cutting stock problem (CSP) relying on the column generation technique to solve the industrial cutting problem, this method known from the modifications that were proposed by (Haessler,1980). Its modifications are based on controlling the generation of cutting patterns by using a more restrictive wording. (Farley, 1988), also proposed improvements of this approach aimed at adapting it to more practical situations. This adaptation is at the level of the generation of an initial solution. (Arbel, 1993) presented a large-scale optimization methods applied to the problem of cutting irregular shapes. Diegel et al (1996) have studied the configuration minimization conditions in the attitude trim loss problem. (Suliman, 2001) developed a procedure for generating cutting patterns based on the creation of a search tree. Umetani et al, (2006) considered a one-dimensional cutting stock problem with a given number of setups, to minimize the number of stock rolls while constraining the number of different cutting patterns within a bound given by users. For this problem, they proposed a local search algorithm that alternately uses two types of local search processes with the 1-add neighborhood and the shift neighborhood, respectively. Delorme et. al,

(2015) reviewed the mathematical models and the exact algorithms developed in the last fifty years for one of the most famous combinatorial optimization problems, the cutting stock problem. They discussed the main approaches proposed in the literature, and they provided an experimental evaluation of the available software on different classes of benchmarks.
Martinovic et al, (2018) considered the one-dimensional cutting stock problem which consists in determining the minimum number of given large stock rolls that has to be cut to satisfy the demands of certain smaller item lengths. They compared the proposed formulas with those in the literature from a theoretical and numerical point of view. Tanir et al, (2019) considered the one-dimensional cutting stock problem with divisible items, which arises in the steel industries. While planning the steel cutting operations, each item can be divided into smaller pieces, and then they can be recombined by welding. The objective is to minimize both the trim loss and the number of the welds. To achieve this they proposed a mathematical model for the problem is given and a dynamic programming based heuristic algorithm.

Within the framework of multi-objective optimization, the first work to appear in the literature was written by Kolen and Spieksma, (2000) on the multi-objective cutting stock problem with cost of setup. A few years later, Golfeto et al, (2009b) developed a symbiotic genetic algorithm applied to the multi-objective cutting stock problem, to minimize the total trim loss and the number of setups. Salles-Neto et al, (2009) presented a study of the conditions for the weakly efficient solutions of the cutting stock problem with two-objectives, in which they considered the cost of trim loss and the cost of setup. Cui and Yang, (2010) considered three objectives: the total panel cost, the profit from the leftovers (the unused length of a panel is leftover once it is longer than a threshold) and the profit from the leftovers coming from past cutting operations. They proposed a two-stage heuristic algorithm: the first stage is a linear program that cuts the major part of the item demand whereas the second stage is a sequential heuristic that cuts the remaining item demand. Araujo et al, (2014) presented a genetic algorithm for the one-dimensional cutting stock problem with setups, considering two conflicting objective functions: minimization of both the number of objects and the number of different cutting patterns used. They proposed a heuristic method based on the concepts of genetic algorithms to solve the problem. Cui et al, (2015) presented a pattern-set generation algorithm for the one-dimensional cutting stock problem with setup cost. Using an integer linear programming model to minimize the sum of material and setup costs over a given pattern set, and then describes a sequential grouping procedure to generate the patterns in the set.

Aliano et al, (2017) studied two objectives: the number of times a cutting pattern is used and the number of different cutting patterns. They aimed to generate all non dominated objective vectors and presented four procedures: the weighted sum method, the Chebyshev's metric, the ε-Constraint method and an improved version of Chebyshev's metric. Campello et al, (2020) presented a multi criteria study for the one dimensional integrated cutting stock and lot sizing problems arising in the paper industry. Their two objectives were minimizing total production costs, inventory costs of paper rolls and setup costs of machines and minimizing total material waste and inventory costs of items. They proposed two solution approaches: the weighting approach that minimizes the weighted sum of the objectives and an ε-constraint method where lot sizing related objective is minimized.

3. Mathematical modeling and methods

Definition 1 *We call a feasible cutting pattern a set of parts cut from an object and the position of each (of them) in the object.*

Definition 2 *We call a feasible cutting plan a set of feasible cutting patterns to satisfy the different types of demand.*

3.1. Mathematical formulation of the **one-dimensional cutting stock problem with setups cost**

In this section, we describe the cutting stock problem with setups cost which was first introduced by (Haessler, 1975). Where we consider a problem which consists in cutting an object of size W into several parts called pieces of sizes $w_i < W$, to satisfy n different sizes that we denote by d_i, i varying from 1 to n (demand vector), and each time we go from one cutting pattern to another, we change the position of the saws in the cutter, this fixing of the cutter often generates significant costs called setup, which we denote by c_2. The number of times the cost c_2 is incurred is equal to the number of cutting patterns used, i.e. the number of variables $x_j > 0$. We introduce a boolean variable $\delta(x_j)$ equal to 1 when $x_j > 0$ is equal to 0 when $x_j = 0$, where j = 1 … T. so by adding the term: $c_2 \sum_{j=1}^{T} \delta(x_j)$ to the total cost to be minimized, the cost function to be optimized is reduced to:

$$\begin{aligned} &Minimizes \quad c_1\left(W \times \sum_{j=1}^{T} x_j - \sum_{i=1}^{n} w_i d_i\right) + c_2 \sum_{j=1}^{T} \delta(x_j) \\ &\qquad s.t. \sum_{j=1}^{T} p_{ij} x_j \geq d_i \qquad i = 1, \dots, n \\ &\qquad x_j \in \mathbb{N} \qquad\qquad j = 1, \dots, T \end{aligned} \tag{1}$$

where c_1 is the unit cost of trim loss, p_{ij} is the frequency of piece i in the cutting pattern j, d_i is the number of type i pieces requested, T is the number of cutting patterns, n is the number of different pieces sizes, x_j is the number of times that the j^{th} cutting pattern is used.

3.2. Multi-objective optimization

The following definitions are according to Ehrgott and Gandibleux, (2000). Multi-objective combinatorial optimization is part of the field of combinatorial optimization. The main specificity of multi-objective being the existence of several functions to be optimized, it is in particular necessary to revisit the notion of optimality of solutions. It can be defined by:

$$\begin{aligned} &Optimize \;\; F(x) = \left(f_1(x), f_2(x), \dots, f_n(x)\right) \\ &\quad s.t. \;\; x \in D. \end{aligned} \tag{2}$$

Where n is the number of objectives ($n \geq 2$), $x = (x_1, x_2 \dots x_k)$ is the vector representing the decision variables, each of the functions f_i is to be optimized i = 1… n, i.e. to say to minimize or maximize and D represents the set of feasible solutions. The set $\mathbb{R}^k$ which contains D is called a decision space. In the following, we suppose that all the objectives are to be minimized. The set $\mathbb{R}^n$ which contains F is called the criteria space or the objective space.

3.2.1. Concept of optimality in the sense of Pareto

In order to compare the solutions in a multi-objective optimization problem, the concept of Pareto dominance is used.

1. Given two vectors of criteria $u, v \in F(D)$. We say that the decision vector u dominates the vector v (denoted $u \leq v$) if and only if:

$$\forall\, i \in \{1, 2, \dots, n\},\; f_i(u) \leq f_i(v) \wedge \exists\, i \in \{1, 2, \dots, k\}\colon\; f_i(u) \prec f_i(v)$$

   (We say that u dominates v in the Pareto sense, means that F(u) is better than F(v) for all objectives, and there is at least one objective function for which F(u) is strictly better than F(v)).

2. A solution $x^* \in D$ is an efficient solution if there is no $x \in D$ such that F(x) dominates $F(x^*)$. Conversely x^* is inefficient. Therefore, a solution $x^*$is efficient if its criterion vector is not dominated by any criterion vector of another solution in D. That is, it is not possible to move in a feasible direction to decrease one of the objectives, without necessarily increasing at least one of the other objective values.

The limiting efficiency is also known as Pareto optimal and the curve in the objective space formed by the non-dominated vectors which are in the Pareto optimal set is called the Pareto front.

3.3. Mathematical formulation of the bi-objective cutting stock problem with setups

Wäscher and Henn, (2013) showed that problem (1) can be replaced by a corresponding bi-objective optimization model, these results in a vector optimization model based on single stock size cutting stock problem inputs with setups. Whereas, the mathematical model of the two-objective cutting stock problem with setup, it will be as follows:

Model Parameters

Let

- W be the length of main object,
- w_i be the length of order piece i, (i = 1,…, n),
- p_{ij} be the number of occurrences of the i^{th} piece in the j^{th} pattern,

Decision variables

Let

- x_j be the number of times the j^{th} cutting pattern is used, j = 1,…, T,

Objective functions

We have two objective functions:

- The total trim loss

$$Min\left(f_1(x)\right) = Min\left(W \times \sum_{j=1}^{T} x_j - \sum_{i=1}^{n} w_i d_i\right) \tag{3}$$

- number of setups

$$Min\left(f_2(x)\right) = Min\left(\sum_{j=1}^{T} \delta(x_j)\right) \tag{4}$$

Within the framework of these notifications, we can formulate a bi-objective model for the two-dimensional cutting stock problem with setup cost described above, in the following form:

$$\begin{aligned}
&Min\left(f_1(x)\right) = Min\left(W \times \textstyle\sum_{j=1}^{T} x_j - \sum_{i=1}^{n} w_i d_i\right)\\
&Min\left(f_2(x)\right) = Min\left(\textstyle\sum_{j=1}^{T} \delta(x_j)\right)\\
&\textstyle\sum_{j=1}^{T} p_{ij} x_j \geq d_i \qquad i = 1, \dots, n\\
&x_j \in \mathbb{N} \qquad j = 1, \dots, T\\
&\delta(x_j) = \begin{cases} 1 \;\; if\; x_j > 0 \\ 0 \;\; if\; x_j = 0 \end{cases}
\end{aligned} \tag{5}$$

3.4. Methods

In this section, we present a technique for generating feasible cutting patterns, as well as the technique of building cutting plans to satisfy all the demands and also we propose a technique to solve the problem.

3.4.1. Generation of feasible cutting patterns

The idea of this heuristic developed here, is that after having arranged the lengths of the pieces types in descending order from the greatest length to the smallest length ($w_1 > w_2 > \ldots > w_n$). At this stage, we calculate the number of times that the length of the first pieces type can be cut from the object length, where we denote it by p_{11}, and also calculate the number of times the length of the remaining piece types can be cut from the rest of the object length. Then, we go down p_{11} by 1, (that is to say, we put $p_{21} = p_{11} - 1$) and we cut the lengths of the rest of the pieces types from the object length. Where we keep shrinking p_{11} each time by 1, until there is no more p_{j1} where j ≥ 1 and in each we cut the lengths of the rest of the pieces types from the object length. Then, we fix p_{j1} in each of the values obtained, and for each value of p_{j1} we go down the number of times that the length of the second type of pieces has been cut from the object length (which means that we put p_{j1} in its first values and we start to lower the value of p_{12} each time by 1, where j ≥ 1) and calculating the number of times that can be cut for the remaining piece types from the object length in the same way as the first. This process is repeated for each part type length until i = n-1. Practically, we proceed as follows:

1. Calculate $p_{11} = \left\lfloor \frac{W}{w_1} \right\rfloor$ where represents ⌊ ⌋ the whole lower part,
2. For j = 1 and i varying from 2 to n, $p_{1i} = \left\lfloor \frac{W - \sum_{z=1}^{i-1} p_{1z} \times w_z}{w_i} \right\rfloor$, (Suliman., 2001),
3. Decrease p_{11} by 1 (i.e. we put $p_{21} = p_{11} - 1$),
4. For j = 2 and i varying from 2 to n, do $p_{2i} = \left\lfloor \frac{W - \sum_{z=1}^{i-1} p_{2z} \times w_z}{w_i} \right\rfloor$,
5. Continue to decrease p_{11} each time by 1 and to calculate p_{ji} until the cancellation of p_{11} where $j > 2$ and i varying from j to n,
6. The algorithm is updated by fixing p_{ji} where j ≥ 1 to the values found previously and the same operations are repeated with the length of the second type of parts,
7. Repeat the same process for each part type length until i = n-1.

In the end, we will have a matrix of size m × n where each of its lines is a cutting pattern that we denote by P and we call it the cutting matrix.

Algorithm 1. Patterns generation algorithm

Input: Standard Length, lengths of the part types,
Output: Feasible patterns,

1. Arrange the lengths of parts requested, w_i, i = 1, 2,…, n in descending order, i.e. ($w_1 > w_2 > \ldots > w_n$), where n is the number of parts requested,
2. Calculate $p_{11} = \left\lfloor \frac{W}{w_1} \right\rfloor$, let be j = 1, for i = 2 to n do $P_{1i} = \left\lfloor \frac{W - \sum_{z=1}^{i-1} p_{1z} \times w_z}{w_i} \right\rfloor$,
3. let be i = 1, h = 1, k = 1
   a) let be j = 1, d = 1,
   b) if $p_{ji} > 0$ then j = j + 1, l = 1,
   c) h = h + 1,
   d) if i = 1 then $p_{hi} = p_{j-1,i} - 1$,
      For i := k + 1 to n do $p_{hi} = \left\lfloor \frac{W - \sum_{z=1}^{i-1} p_{hz} \times w_z}{w_i} \right\rfloor$,
   e) else for z := 1 to i-1 do $p_{hz} = p_{j-1,z}$,
      For i := k do $p_{hi} = p_{j-1,i} - 1$,
      For i: = k + 1 to n do $p_{hi} = \left\lfloor \frac{W - \sum_{z=1}^{i-1} p_{hz} \times w_z}{w_i} \right\rfloor$,
   f) if $p_{hi} > 0$ then let be l = l + 1, d = d + 1 and go to (c) else go to (g),
   g) d = d + 1,
   h) if $d < m$, then go to (b) else if $i < n - 1$ then let be i = i + 1, k = k + 1 and go to (a), else stop,
4. Else let be j = j + 1, d = d + 1 and go to (b).

3.4.2. Cutting plans construction strategy

In this section, we will show how to construct cutting plans and to justify this we present the following proposal.

Proposition. *Let P be a m ×n boolean matrix, w a column n-vector with strictly positive integer coordinates and a m-vector column is given by V = P × w.*
Let r, i, j, k and s be integers such that $1 \le r, i, j, k, s \le m$ with $P_s' = P_i + P_j + \cdots + P_k$ a sum of line vectors in the matrix P (respectively P_r a line vector in the matrix P) and $v_s' = v_i + v_j + \cdots + v_k$, (∀ i ≠ j ≠ k) a sum of components in the vector V corresponding to the vector P_s' (respectively v_r a component in the vector V corresponding to the vector P_r), we have the following proposition: $v_s' \ge \sum_{r=1}^{n} w_r \Leftrightarrow P_s' \ge (1, \dots, 1)$.

Proof

(i) If $v_s' \ge \sum_{r=1}^{n} w_r$ then $P_s' \ge (1, \dots, 1)$, by opposite we have: if $P_s' < (1, \dots, 1)$ then $v_s' < \sum_{r=1}^{n} w_r$. We multiply on the right the inequality $P_s' < (1, \dots, 1)$ by the column vector w, therefore: $P_s' w < (1, \dots, 1)w \Leftrightarrow P_s' w < (w_1 + w_2 + \cdots + w_n) \Leftrightarrow v_s' < \sum_{r=1}^{n} w_r$.

(ii) If $P_s' \ge (1, \dots, 1)$, then $v_s' \ge \sum_{r=1}^{n} w_r$. We have: $P_s' \ge (1, \dots, 1) \Leftrightarrow P_{s1}' \ge 1, P_{s2}' \ge 1, \dots, P_{sn}' \ge 1$. We multiply on the right each inequality of $P_{si}' \ge 1$ by w_i where $w_i \in \mathbb{N}^*$and $1 \le i \le n$, hence: $P_{s1}' w_1 \ge w_1, \dots, P_{sn}' w_n \ge w_n \Leftrightarrow P_{s1}' w_1 +, \dots, + P_{sn}' w_n \ge w_1 +, \dots, + w_n \Leftrightarrow P_s' w \ge \sum_{r=1}^{n} w_r \Leftrightarrow v_s' \ge \sum_{r=1}^{n} w_r$. □

The process of building the cutting planes depends mainly on the cutting patterns that were generated by the previous algorithm, because each subset of these patterns satisfying the demands is considered as a cutting plane and to form these planes we code each feasible cutting pattern, where we assign the number 1 to each component p_{ij} different from 0, and we assign the number 0 to each component p_{ij} equal to 0. We therefore form a coded cutting pattern; each of its values is either 0 or 1. Thus, a coded matrix is formed denoted by P'. Where, we multiply P' by the sizes of the requested part types w, we get a column vector denoted by V, where $V = \sum_{r=1}^{m} \sum_{j=1}^{n} p_{rj}' w_j$ and according to the previous proposition, we have for each sum of the components $v_s' = v_i + v_j + \cdots + v_k$, greater than or equal to $\sum_{r=1}^{n} w_r$ and $P_s' = P_i' + P_j' + \cdots + P_k' \ge (1, \dots, 1)$, all the cutting patterns P_i', P_j', P_k' corresponds to these components, forms a cutting plane (denoted by pd_i) and thus, if there exists in vector V, a component v_r greater than or equal to $\sum_{r=1}^{n} w_r$, and $P_r' \ge (1, \dots, 1)$ then the cutting pattern P_r' corresponds to this component forms a cutting plane and therefore each cutting plane with the smallest trims loss and a fixed number of setups, is an efficient solution.

3.3.3. Problem solving

Problem solving goes through two stages, the modeling stage and the resolution stage.

Step 1. At this stage, we model how to calculate the sums of the components of vector V, that is, we model the v_s', where $v_s' = v_i + v_j + \cdots + v_k$, by a graph G = (X, U, C), where X : the set of vertices, U: the set of arcs, C: the evaluation of the arcs, in which, each component of the vector V represents by a vertex x of G and each sum of components $x_i + x_j$ described by a arc u connect the vertex x_i by the vertex x_j, we also add a source vertex r join the vertices x of G which are located in front of the source vertex r. Then, we continue the modeling process as follows:

1. For j = 1 and i varying from 1 to m, let be $x_{ij} = x_i$,
2. For j varying from 1 to m-2 and i varying from m - j + 1 to 2 let be $x_{i-1,j+1} = x_{ij}$,
3. j varying from 1 to m-1 and i varying from 1 to m - j we connect the vertices x_{ij} by the vertices $x_{i+1,j}$,
4. For j varying from 1 to m-2, i varying from 1 to m - j – 1 and z varying from i + 1 to m we connect the vertices x_{ij} by the vertices $x_{z,j+1}$.

Step 2. In this next step, we are researching a solution to the problem, which consists in finding the set of efficient solutions. In the graph that was formed from the modeling phase, we first start to initialize E = {r}, E_{1j} = Ø, (j = 1, 2…) and we fix $c\big(I(u)\big) = 0$ for each arc u at an initial end the vertex r. Then, we determine the vertices x_{ij} in X - E whose predecessors are in E, and we let be $E_{ij} = E_{ij} \cup \{x_{ij}\}$, as we calculate $S\big(T(u)\big) = c\big(I(u)\big) + c(u)$. Now we do the test, if $S(T(u)) \geq \sum_{r=1}^{n} w_r$ and $P_s' = P_i' + P_j' + \cdots + P_k' \geq (1, \dots, 1)$, where P_i', P_j', P_k' the cutting patterns correspond to v_i, v_j, v_k, then the cutting plane pd_i corresponds to $c\big(I(u)\big) = S(T(u))$ is a feasible solution. In the next stage, we let be $c\big(I(u)\big) = S(T(u))$ for the arcs which arec adjacent to the same vertex already defined as a predecessor. Then, we determine the vertices x_{ij} in X - E whose predecessors are in E. Let's repeat the same steps as the first stage. Where the calculation procedure is continued in the same way until we get very close (not improving the solution) or reach the theoretical value $\sum_{j=1}^{T} x_j = \left\lceil \frac{\sum_{i=1}^{n} w_i d_i}{W} \right\rceil$, where $\lceil\ \rceil$ the upper integer part.

Algorithm 2. Cutting plan construction algorithm

Input: Feasible cutting patterns,
Output: Efficient solutions,

1. Apply the patterns generation algorithm,
2. Calculate the theoretical value: $\sum_{j=1}^{T} X_j = \left\lceil \frac{\sum_{i=1}^{n} l_i \times w_i \times d_i}{W} \right\rceil$,
   a) If there are cutting patterns P_r where $P_r \geq (1, \dots, 1)$, then determine the corresponding cutting planes pd_i and go to (b), else go to (3),
   b) Calculate $x_j = \mathrm{Max}\left(\left\lceil \frac{d_i}{p_{ij}} \right\rceil\right)$, whrer $\left\lceil \frac{d_i}{p_{ij}} \right\rceil$, the upper integer part of $\frac{d_i}{p_{ij}}$ and $P_{ij} \neq 0$,
   c) Calculate the trims loss $pt_i = W \times \sum_{j=1}^{T} x_j - \sum_{r=1}^{n} w_r d_r$,
   d) Calculate Pt = Min(pt_i), and go to (3),
3. Code the feasible cutting patterns P_i as follows: $P_{ij}' = \begin{cases} 1 \text{ if } p_{ij} \neq 0 \\ 0 \text{ if } p_{ij} = 0 \end{cases}$
4. Calculate $V = \sum_{r=1}^{m} \sum_{j=1}^{n} p_{rj}' w_j$,
5. Let be j = 1, for i := 1 to m do $x_{ij} = x_i$,
6. For j := 1 to m-2 do
   For i : = m - j + 1 to 2 do $x_{i-1,j+1} = x_{ij}$,
7. For j := 1 to m-1 do,
   For i : = 1 to m - j connect the vertex x_{ij} by the vertex $x_{i+1,j}$,
8. For j := 1 to m - 2 do
   For i := 1 to m - j - 1 do
   For z : = i + 1 to m - j join x_{ij} by $x_{z,j+1}$,
9. Let be E = {r}, E_{1j} = Ø, j = 1, 2…, c(I(u)) = 0 for each arc u at an initial end the vertex r, and go to (10),
10. Determine the vertices x_{ij} in X - E whose predecessors are in E and let be $E_{ij} = E_{ij} \cup \{x_{ij}\}$,
11. Calculate $S\big(T(u)\big) = c(I(u)) + c(u)$, where u a terminal end arc(T) is the vertex x_{ij},

a) If $S(T(u)) \geq \sum_{r=1}^{n} w_r$ then determine the cutting patterns corresponding to the summits adding up $S(T(u))$ and go to (b), else go to (g),
b) If $P'_s = P'_i + P'_j + \cdots + P'_k \geq (1, \dots, 1)$, then determine the cutting plane pd_i corresponds to the decoded patterns P_i, P_j, P_k, and go to steps (c), (d), (e) and (f) else go to (g),
c) Calculate $b_{ij} = \sum_{i\geq 1} p_{ij}$, x_j = $Max\left(\left\lceil \frac{d_i}{b_{ij}} \right\rceil\right)$ where $p_{ij} \neq 0$, and calculate $x_{pd_i} = \sum_{j=1}^{T} x_j$,
d) If there are cutting plans for the same number of setups and the same x_{pd_i}, eliminate the redundant plans and go to (e), else go to (e),
e) Calculate the trim loss: $pt_i = W \times \sum_{j=1}^{T} x_j - \sum_{r=1}^{n} w_r d_r$,
f) Calculate pt = Min (pt_i),
g) Let be $c(I(u)) = S(T(u))$ for arcs which are adjacent to the same vertex already defined as a predecessor,
h) If the number of arcs entering at a T(u) is greater than 1 then burst this end to the number of arcs entering at T(u), then let be $c(I(u')) = S(T(u'))$, $c(I^{'}(u'')) = S(T(u''))$, …, and go to (i), else go to (i),
i) Repeat $E = E \cup \{x_{ij}\}$, and go to (10) until reaching or approaching the theoretical value of $\sum_{j=1}^{T} X_j = \left\lceil \frac{\sum_{i=1}^{n} l_i \times w_i \times d_i}{W} \right\rceil$ j = 1, 2….

3.4.4. Finiteness of the algorithm

We check in this paragraph that the number of iterations is finite and the algorithm does not loop:

(i) Cutting pattern generation algorithm: The idea in this step consists in calculating the frequency p_{11} of the first part w_1 on the main object W and each time decreased this frequency by 1 until the cancellation of p_{11} and calculate the other frequencies by $p_{1i} = \left\lfloor \frac{W - \sum_{z=1}^{i-1} p_{1z} \times w_z}{w_i} \right\rfloor$, the algorithm reiterated for each part until i = n-1. Indeed a considered part is not revisited a second time, so the algorithm does not loop and as the number of parts is finite then the number of iterations are finite, as shown in the following flowchart:

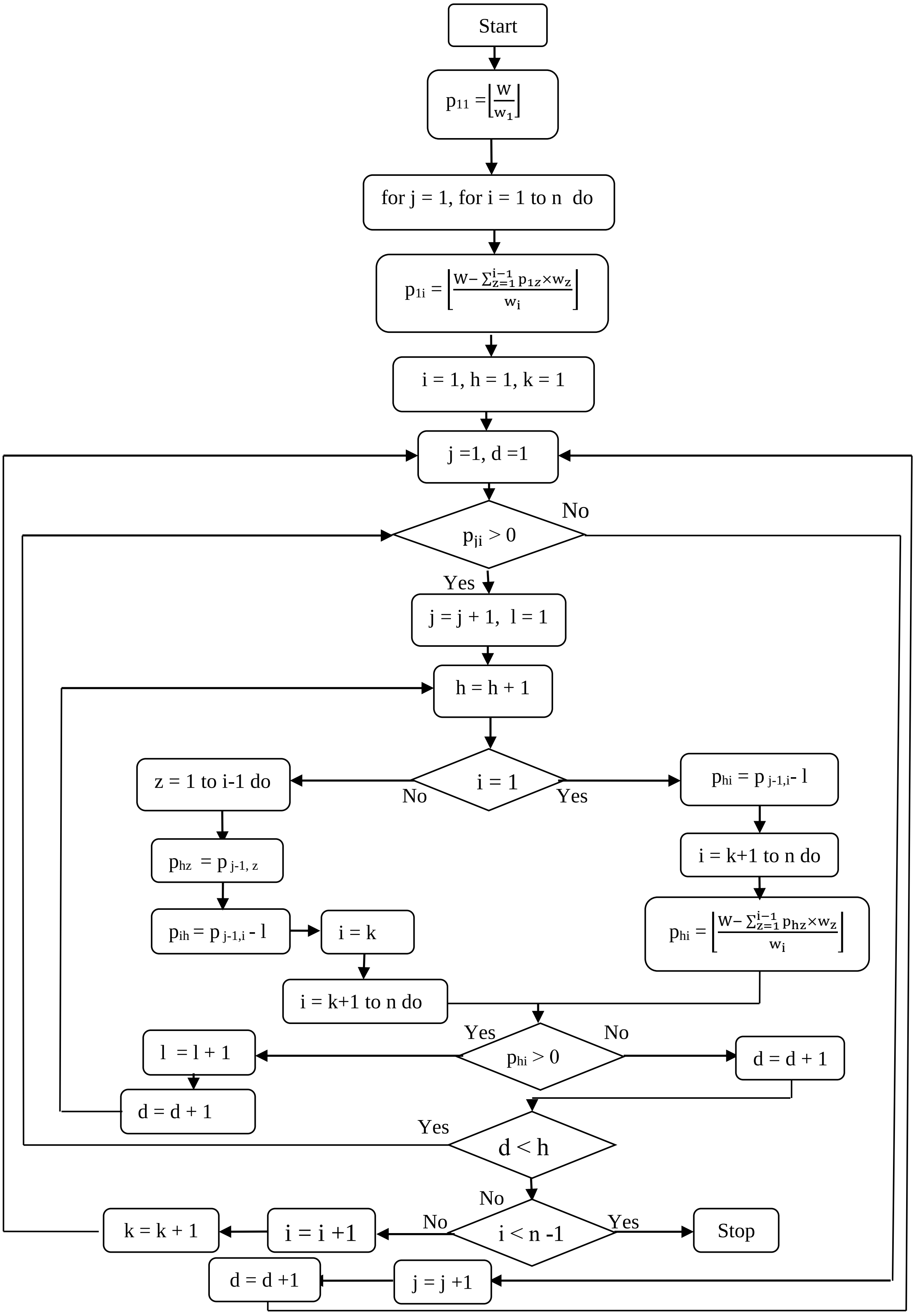
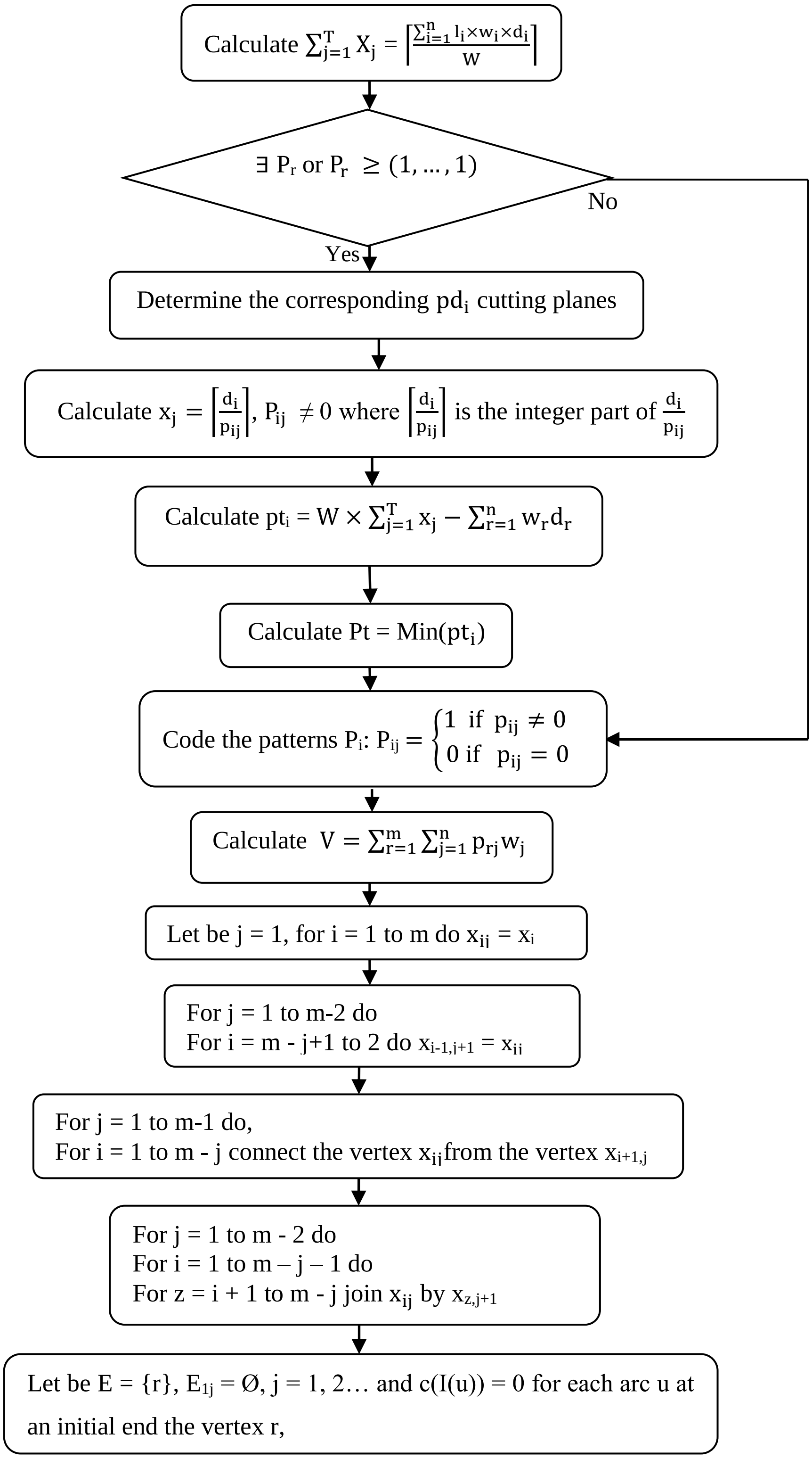


Figure 1. Flowchart of the proposed algorithm

(ii) Construction of the cutting planes, in this step the algorithm consists in adding the evaluations of the arcs of all the possible elementary paths in the graph G = (X, U, C), close to close, from the source vertex to where we can't develop the solution, in fact, only a vertex considered is visited only once, and since G is without circuit, therefore we do not loop in the algorithm, and as G with a finite number of vertices and arcs then the number of iterations is finite, as shown in the following flowchart:



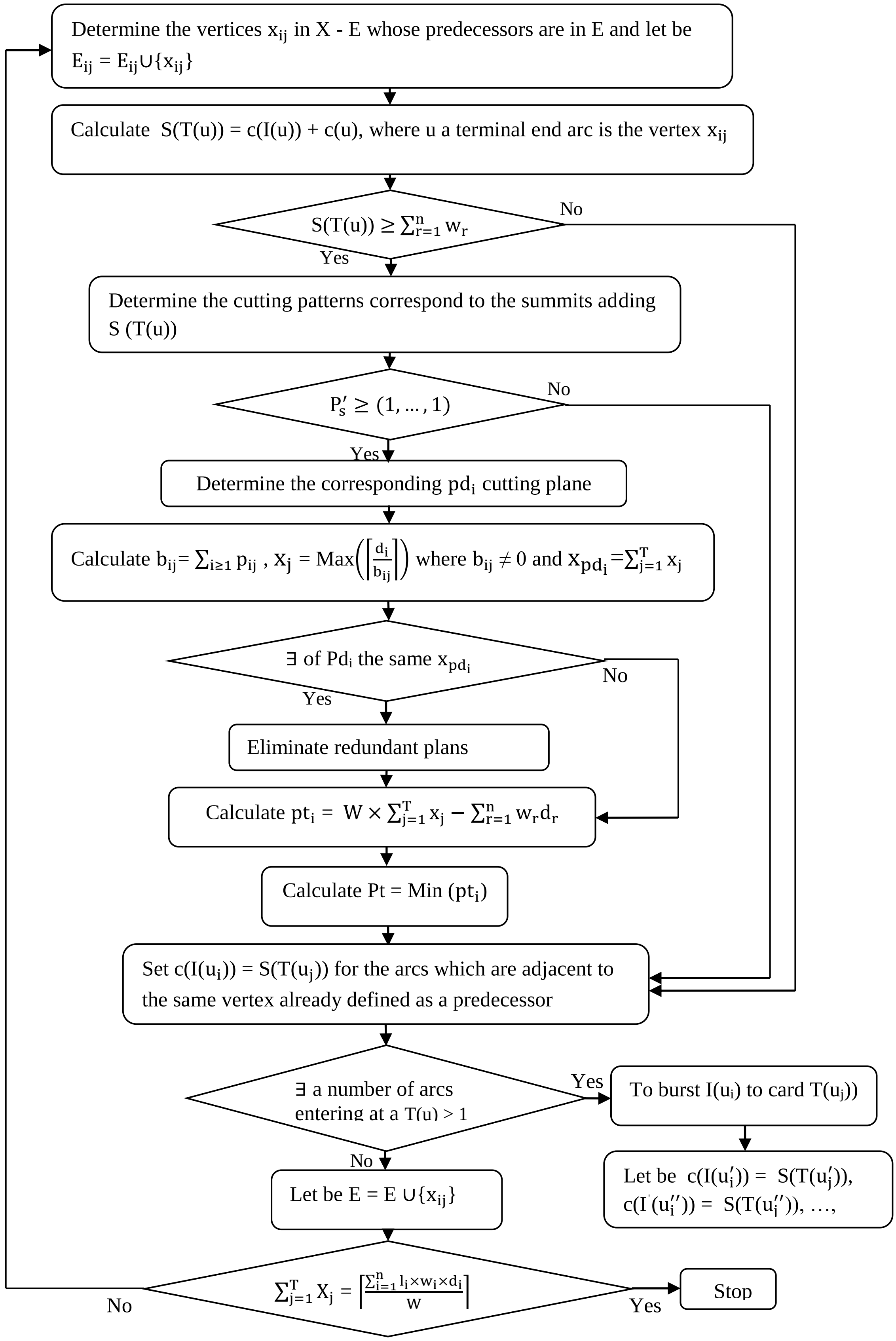


Figure 2. Flowchart of the proposed algorithm

3.4.5. Illustrative example.

In this example we run the proposed algorithm.

• W = 80, n = 3,

• w = (w_1, w_2, w_3) = (50, 40, 30),

• d = (d_1, d_2, d_3) = (4, 6, 11),

1. Calculate $p_{11} = \left\lfloor \frac{80}{50} \right\rfloor = 1$, Let be j = 1, for i = 2 to 3 do $p_{12} = \left\lfloor \frac{W - \sum_{z=1}^{1} p_{1z}}{w_2} \right\rfloor = \left\lfloor \frac{80 - 1 \times 50}{40} \right\rfloor = 0$
   $p_{13} = \left\lfloor \frac{80 - 1 \times 50 - 0 \times 40}{30} \right\rfloor = 1,$
   P = (1 0 1)
2. Let be i = 1, h =1, k =1,

a) Let be j =1, d =1,

b) $p_{11} = 1 > 0$ then j := j +1, l =1,

c) h = h + 1= 1 + 1 = 2,

d) i = 1 then $p_{21} = p_{11} - 1 = 1 - 1 = 0$,
For i = 2 to 3 do $p_{22} = \left\lfloor \frac{80 - 0 \times 50}{40} \right\rfloor = 2$, $p_{23} = \left\lfloor \frac{80 - 0 \times 50 - 1 \times 40}{30} \right\rfloor = 1$,
$P = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \end{pmatrix}$

e) p_{21} = 0 then go to (f),

f) d = d + 1 = 1 + 1 = 2,

g) d = m = 2

h) i = 1< n -1 = 2 then let be i = i + 1= 2, k = k +1= 2, and go to (a),

a) Let be j = 1, d = 1,

b) p_{12}= 0,

i) Let be j = j +1= 2, d = d +1=2,

b) p_{22} = 2 > 0 then j = j + 1= 3, l = 1,

c) h = h + 1= 2 + 1= 3,

d) For z = 1 to 1 do $p_{31} = p_{21}$ = 0,
For i = k = 2 do $p_{32} = p_{22} - 1 = 2 - 1 = 1$,
For i = 3 to 3 do $p_{33} = \left\lfloor \frac{80 - 0 \times 50 - 1 \times 40}{30} \right\rfloor = 1$,
$$P = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix},$$

e) P_{32} = 1 > 0, then let be l = l + 1= 2, d = d + 1 = 3 go to (c),

c) h = h + 1 = 4,

d) For z = 1 to 1 do $p_{41} = p_{31}$ = 0,
For i = k = 2 do $p_{42} = p_{22}$ -2= 2 – 2= 0,
For i = 3 to 3 do $p_{43} = \left\lfloor \frac{80 - 0 \times 50 - 0 \times 40}{30} \right\rfloor = 2$,
$$P = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix},$$

e) P_{42} = 0, d = d + 1= 4, i = 2 = n -1, stop.
$\nexists$ a cutting pattern P_r, (r = 1,…, 4) where $P_{rj} \neq 0$, (j = 1,…, 3), then go to (4),

3. The theoretical value east $\sum_{j=1}^{T} X_j = \left\lceil \frac{770}{80} \right\rceil$ = 10,
4. The feasible cutting patterns coded are shown in the following matrix:

$$P' = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

5. The vector $V = \sum_{r=1}^{m}\sum_{j=1}^{n} p'_{ij}w_j$, presented in the following vector: $V = \begin{pmatrix} 80 \\ 40 \\ 70 \\ 30 \end{pmatrix}$

Be $X = (x_1, x_2, x_3, x_4)$, $U = (u_1, u_2, u_3 \ ...)$, C = (80, 40, 70, 30) et m = cardinal (V) = 4.
j = 1, for i = 1 to 4 do $x_{11} = x_1$, $x_{21} = x_2$, $x_{31} = x_3$, $x_{41} = x_4$,

1. For j = 1 to 2, for i = 4 – j + 1 to 2 do $x_{i-1,j+1} = x_{ij}$,
   j = 1, for i = 4 to 2 do $x_{32} = x_{41}$, $x_{22} = x_{31}$, $x_{12} = x_{21}$,
   j = 2, for i = 3 to 2 do $x_{23} = x_{32}$, $x_{13} = x_{23}$,
6. For j = 1 to 3, for i = 1 to 4 – j connect the vertex x_{ij} by the vertex $x_{i+1,j}$,
   For j = 1, for i = 1 to 3, connect x_{11} by x_{21}, connect x_{21} by x_{31}, connect x_{31} by x_{41}
   For j = 2, for i = 1 to 2, connect x_{12} by x_{22}, connect x_{22} by x_{32},
   For j = 3, for i = 1, connect x_{13} by x_{23},
7. For j = 1 to 2, for i = 1 to 4 – j – 1, for z = i + 1 to 4 - j join x_{ij} by $x_{z,j+1}$,
   j = 1, i = 1, for z = 2 to 3, join x_{11} by x_{22}, join x_{11} by x_{32},
   j = 1, i = 2, for z = 3, join x_{21} by x_{32},
   j = 2, i = 1, for z = 2, join x_{12} by x_{23},

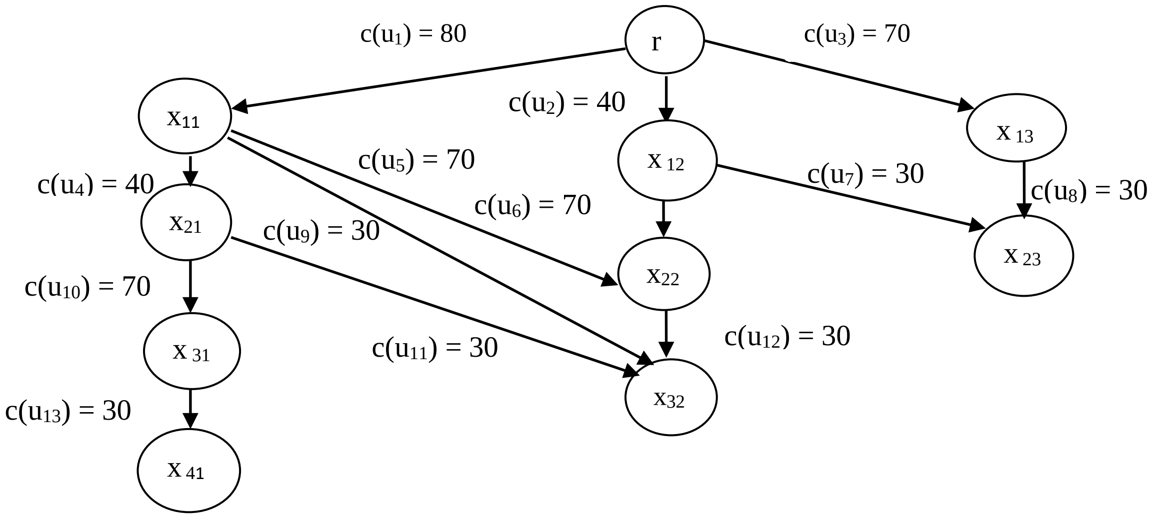


Figure 3. Sum of vector components graph V

E = {r}, $E_{1j} = \emptyset$, (j = 1… 3), $c(I(u_1)) = 0$, $c(I(u_2)) = 0$, $c(I(u_3)) = 0$,

1st iteration

10. The vertex x_{11}, x_{12}, x_{13} are vertices in X – E whose predecessors are in E, Let be $E_{11} = \{x_{11}\}$, $E_{12}=\{x_{12}\}$, $E_{13}=\{x_{13}\}$,
11. $S(T(u_1)) = c(I(u_1)) + c(u_1)) = 0 + 80 = 80 < \sum_{r=1}^{3} w_r$, $S(T(u_2)) = c(I(u_2)) + c(u_2) = 0 + 40 < \sum_{r=1}^{3} w_r$, $S(T(u_3)) = c(I(u_3)) + c(u_3) = 0 + 30 = 30 < \sum_{r=1}^{3} w_r$, go to(c).

c) Let be $c(I(u_4)) = S(T(u_1))$, $c(I(u_5)) = S(T(u_1))$, $c(I(u_9)) = S(T(u_1))$, $c(I(u_6)) = S(T(u_2))$, $c(I(u_7)) = S(T(u_2))$, $c(I(u_8)) = S(T(u_3))$, and go to (d),

d) Let be E = {r} ∪ $\{x_{11}, x_{12}, x_{13}\}$ = $\{r, x_{11}, x_{12}, x_{13}\}$ and go to (10),

10. The vertex x_{21}, x_{22}, x_{23} are vertices in X – E whose predecessors are in E, Let be $E_{21} = \{x_{11}, x_{21}\}$, $E_{22} = \{x_{12}, x_{22}\}$, $E_{23} = \{x_{13}, x_{23}\}$,

11. $S(T(u_4)) = c(I(u_4)) + c(u_4) = 80 + 40 = 120$, $S(T(u_5)) = c(I(u_5)) + c(u_5) = 80 + 70 = 150$, $S(T(u_6)) = c(I(u_6)) + c(u_6) = 40 + 70 = 110$, $S(T(u_7)) = c(I(u_7)) + c(u_7) = 40 + 30 = 70$, $S(T(u_8)) = c(I(u_8)) + c(u_8) = 70 + 30 = 100$.

a) $S(T(u_4)) \geq \sum_{r=1}^{3} w_r = 120$ so the cutting patterns $P_1' = (1, 0, 1)$ et $P_2' = (0, 1, 0)$ corresponds to v_1, v_2, can be formed a cutting plane, go to (b),

b) $P_1'' = P_1' + P_2' = (1, 1, 1) \geq (1, 1, 1)$ then the decoded cutting patterns P_1, P_2 forms a noted cutting plane $pd_1 = \begin{cases} P_1 = (1,0,1) \\ P_2 = (0,2,0) \end{cases}$,

   a) $S(T(u_5)) \geq \sum_{r=1}^{3} w_r = 120$ so the cutting patterns $P_1' = (1, 0, 1)$ and $P_3' = (0, 1, 1)$, corresponds to v_1, v_3, can be formed a cutting plane, go to (b),

   b) $P_2'' = P_1' + P_3' = (1, 1, 2) \geq (1, 1, 1)$ then the decoded cutting patterns P_1, P_3 forms a noted cutting plane $Pd_2 = \begin{cases} P_1 = (1,0,1) \\ P_3 = (0,1,1) \end{cases}$, and go to (c).

   c) $x_{pd_1} = \begin{cases} \text{Max}\left\lceil \frac{4}{1}, \frac{11}{1} \right\rceil = 11 \\ \text{Max}\left\lceil \frac{6}{2} \right\rceil = 3 \end{cases}$, $x_{pd_2} = \begin{cases} \text{Max}\left\lceil \frac{4}{1}, \frac{11}{2} \right\rceil = 6 \\ \text{Max}\left\lceil \frac{6}{1}, \frac{11}{2} \right\rceil = 6 \end{cases}$,

   $pt_{pd_1} = 80 \times (11 + 3) - (4 \times 50 + 6 \times 40 + 11 \times 30) = 3, 5$ %,

   $pt_{pd_2} = 80 \times (6 + 6) - (4 \times 50 + 6 \times 40 + 11 \times 30) = 1, 9$ %,

   $Pt = \text{Min}(pt_{pd_1}, pt_{pd_2}) = 1.9\%$,

   d) Let be $c(I(u_{10})) = S(T(u_4))$, $c(I(u_9)) = S(T(u_1))$, $c(I(u_{11})) = S(T(u_4))$, $c(I(u_{12}')) = S(T(u_5))$, $c(I(u_{12}'')) = S(T(u_6))$ and go to (e),

   e) Let be $E = \{x_{11}, x_{12}, x_{13}\} \cup \{x_{21}, x_{22}, x_{23}\} = \{r, x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}\}$ and go to (10),

2^nd^ iteration

10. The vertices x_{31}, x_{32} are vertices in X – E whose predecessors are in E, $E_{31} = \{x_{11}, x_{21}, x_{31}\}$, $E_{32} = \{x_{11}\ x_{21}, x_{32}\}$, $E_{32}' = \{x_{11}, x_{22}, x_{32}'\}$, $E_{32}'' = \{x_{12}, x_{22}, x_{32}''\}$,

11. $S(T(u_{10})) = c(I(u_{10})) + c(u_{10}) = 120 + 70 = 190$, $S(T(u_9)) = c(I(u_9)) + c(u_9) = 80 + 30 = 110$, $S(T(u_{11})) = c(I(u_{11})) + c(u_{11}) = 120 + 30 = 150$, $S(T(u_{12}')) = c(I(u_{12}')) + c(u_{12}') = 150 + 40 = 190$, $S(T(u_{12}'')) = c(I(u_{12}'')) + c(u_{12}'') = 110 + 30 = 140$.

a) $S(T(u_{10})) \geq \sum_{r=1}^{3} w_r = 120$ so the cutting patterns $P_1' = (1, 0, 1)$, $P_2' = (0, 1, 0)$ and $P_3' = (0, 1, 1)$, corresponds to v_1, v_2, v_3, can be formed as a cutting plane, go to (b),

b) $P_3'' = P_1' + P_2' + P_3' = (1, 2, 2) \geq (1, 1, 1)$ then the decoded cutting patterns P_1, P_2 and P_3, forms a noted cutting plane $Pd_3 = \begin{cases} P_1 = (1,0,1) \\ P_2 = (0,2,0), \\ P_3 = (0,1,1) \end{cases}$

a) $S(T(u_{11})) \geq \sum_{r=1}^{3} w_r = 150$, so cutting patterns $P_1' = (1, 0, 1)$, $P_2' = (0, 1, 0)$ and $P_4' = (0, 0, 1)$, corresponds to v_1, v_2, v_4, can be formed a cutting plane, go to (b),

b) $P_4'' = P_1' + P_2' + P_4' = (1, 1, 2) \geq (1, 1, 1)$ then the decoded cutting patterns P_1, P_2 and P_4, forms a noted cutting plane $Pd_4 = \begin{cases} P_1 = (1,0,1) \\ P_2 = (0,2,0) \\ P_4 = (0,0,2) \end{cases}$,

a) $S(T(u_{12}')) \geq \sum_{r=1}^{3} w_r = 120$ so cutting patterns $P_1' = (1, 0, 1)$, $P_3' = (0, 1, 1)$ and $P_4' = (0, 0, 1)$, corresponds to v_1, v_3, v_4, can be formed a cutting plane, go to (b),

b) $P_5'' = P_1' + P_3' + P_4' = (1, 1, 3) \geq (1, 1, 1)$ then the decoded cutting patterns P_1, P_3 and P_4, forms a noted cutting plane $Pd_5 = \begin{cases} P_1 = (1,0,1) \\ P_3 = (0,1,1) \\ P_4 = (0,0,2) \end{cases}$

a) $S(T(u''_{12})) \geq \sum_{r=1}^{3} w_r = 120$ so cutting patterns P'_2 = (0, 1, 0), P'_3 = (0, 1, 1) and P'_4 = (0, 0, 1) corresponds respectively to v_2, v_3, and v_4, can be formed as a cutting plane, go to (b),

b) $P''_6 = P'_2 + P'_3 + P'_4 = (P'_{21} + P'_{31} + P'_{41}, P'_{22} + P'_{32} + P'_{42}, P'_{23} + P'_{33} + P'_{43}) = (0, 2, 1)\ \exists\ P'_{21} + P'_{31} + P'_{41} = 0$, then $\nexists\ pd_i$, go to (c),

c) $x_{pd_3} = \begin{cases} \text{Max}\left[\frac{4}{1}, \frac{11}{2}\right] = 6 \\ \text{Max}\left[\frac{6}{3}\right] = 2 \\ \text{Max}\left[\frac{6}{3}, \frac{11}{2}\right] = 6 \end{cases}$, $x_{pd_4} = \begin{cases} \text{Max}\left[\frac{4}{1}, \frac{11}{3}\right] = 4 \\ \text{Max}\left[\frac{6}{2}\right] = 3 \\ \text{Max}\left[\frac{11}{3}\right] = 4 \end{cases}$, $x_{pd_5} = \begin{cases} \text{Max}\left[\frac{4}{1}, \frac{11}{4}\right] = 4 \\ \text{Max}\left[\frac{6}{1}, \frac{11}{4}\right] = 6 \\ \text{Max}\left[\frac{11}{4}\right] = 3 \end{cases}$

d) pt_{pd_3}= 80 × (6 + 6 + 2) – (7 × 50 + 11 ×40 + 4×30) = 3, 5 %,
pt_{pd_4}= 80 × (4 + 3 + 4) – (4 × 50 + 6 ×40 + 11×30) = 1, 1 %,
pt_{pd_5}= 80 × (4 + 6 + 3) – (7 × 50 + 11 ×40 + 4×30) = 2, 7 %,
Pt = Min($pt_{pd_3}, pt_{pd_4}, pt_{pd_5}$) = 1, 1 % .

Stop because the value of $\sum_{j=1}^{3} X_j$= 4 + 3 + 4 = 11, is very close to the theoretical value in the sense of the integer number $\sum_{j=1}^{T} X_j$ = 10, therefore is not improvement of solution.

So the set of efficient solutions presented in the following table:

Table 1. Set of efficient solutions

Nº of Solutions	Efficient solutions	Percent (%) of Trims loss	Number of setups
1	Pd_4	1.1	3
2	Pd_1	1.9	2

4. Results and discussion

4.1. Results

In the first part of this section, we present the results obtained using our proposed method called construction of the cutting plan method noted by (CCPM) and compare them with some of the results found in the literature.

Where we first compare it with a technique for solving a one-dimensional single-objective cutting stock problem, we have chosen for this a work published by Pazand and Mohammadi, (2009) where they proposed an extension of Haessler's heuristic algorithm to solve a one-dimensional cutting stock problem with a cost of setups, (We noted it by EHH) and they treated real examples in the film industry (from our comparative study we use examples 1 and 2) by three different methods:

1. Integer linear programming method of Gilmore and Gomory (ILP),
2. Sequential heuristic method of Haessler (SHH),
3. Hybrid method of Pazand and Mohammadi (EHH).

In the second part of this section, we will compare the CCPM technique to that of the genetic algorithm proposed by Araujo et al. (2014) to solve a one-dimensional bi-objective cutting stock problem with setups, where we select 30 instances out of 40 available in Umetani et al. (2003), as these examples are taken from a chemical fiber company in Japan. According to the following data:

- Number of types of ordered items: n varies from 4 to 20,
- Object length*:* W = 5180 for the first 15 instances and W = 9080 for the remaining 15 instances,
- Item length: w_i were randomly generated in the interval [500, 2000],

- Demand: d_i were randomly generated in the interval [2, 264] and they aggregated the demand d_i of the items that have the same length w_i. Which is common in the literature (Lee, 2007), Golfeto et al. (2009b).

We present the data of the first example in Table 2, as for the results obtained by the three methods in Table 3, and the set of efficient solutions obtained using CCPM is presented in Table 4. And for example 2, we present the data in Table 5. As for the results obtained by the three methods in Table 6 and the set of efficient solutions obtained using CCPM is presented in Table 7.

Table 2. Required cutting lengths and demands

n =13			W = 6480 mm		
i	w_i (mm)	d_i (mm)	i	w_i (mm)	d_i (mm)
1	1200	14	8	720	62
2	1100	8	9	680	24
3	1020	22	10	600	72
4	960	14	11	560	8
5	900	8	12	510	100
6	850	8	13	400	12
7	760	40			

Table 3. Results obtained by the three methods

Method	Percent (%) of Trims loss	Number of cutting patterns
ILP	0	11
SHH	18.4	6
EHH	5.3	6

Table 4. Result obtained by our algorithm (CCPM)

N° efficient solutions	Percent (%) of Trims loss	Number of setups
1	2.13	13
2	2.86	12
3	3.67	11
4	4.52	10
5	4.61	9
6	5.24	8
7	6.73	7
8	7.31	6
9	8.46	5

Table 5. Required cutting lengths and demands

n =13			W = 6480 mm		
i	w_i (mm)	d_i (mm)	i	w_i (mm)	d_i (mm)
1	510	54	8	950	12
2	600	18	9	1020	32
3	680	61	10	1100	30
4	720	17	11	1140	22
5	730	33	12	1200	32
6	760	14	13	1356	54
7	900	12			

Table 2 represents a vector of efficient solutions for an instance of size n = 13, for different part type lengths and different order sizes. The accuracy of the results obtained ranges from 8.46% total trim loss and 5 setups to 2.13% total trim loss and 13 setups, with an estimated execution time of less than one minute. However, compared to the results in Table 3, there are techniques used that gave a better result.

The results obtained by the three methods are shown in the following table:

Table 6. Results obtained by the three methods

Method	Percent (%) of Trims loss	Number of cutting patterns
ILP	0.36	13 (one of them infeasible)
SHH	6.62	8
EHH	9.98	5

Table 7. Result obtained by CCPM

N° efficient solutions	Percent (%) of Trims loss	Number of setups
1	1.51	13
2	1.90	12
3	2.07	11
4	3.61	10
5	3.89	9
6	4.02	8
7	4.21	7
8	4.72	6
9	5.01	5
10	5.20	4

The results obtained by the proposed method by Araujo et al. (2014) named MOGA and our method named CCPM are shown in Table 8 for the object length W = 5180 and Table 9 for the object length W = 9080.

The results obtained by CCPM in this example represent a range of efficient solutions ranging from 5.20% trim loss of raw material with 4 numbers of setups to 1.51% of total trim loss with 13 numbers of setups. Comparison of these results with those given in Table 6 shows that CCPM dominates EHH because the latter provides an optimal solution of 9.98% wasted raw materials and 5 numbers of setups, while CCPM in 5 setups the total trim loss is 5.01%. Same for SHH, CCPM provides an efficient solution of 4.02% wasted raw materials with 8 setups, while SHH provides an optimal solution of 6.62% wasted raw materials at the same number of setups. As for ILP, it was better than CCPM as it provides a solution 0.36% wasted raw materials with 13 setups, while CCPM with the same number of setups has wastage of 1.51%.

Reading the results presented in Table 8, we can see our proposed method provided efficient solutions for each of the instance studied, so that each solution is characterized by a number of setups and a quantity of raw material waste of a reasonable execution time, where it did not exceed in most cases a minute.

Comparing these results is with those obtained by MOGA, where it was noted that CCPM is dominant in the majority of instances, for example, in instances of Fiber 06, Fiber 07, Fiber 09, Fiber 10, Fiber 13b, Fiber 17 and Fiber 28 and other. CCPM dominates MOGA, while the latter dominates in the solution of Fiber 08, Fiber 13a, and Fiber 19.

In Table 9, the length of the main object has been increased while the lengths of the requested part types remain the same; we have observed that the CCPM has remained stable for this change, which means that it has provided solutions for all instances studied with the same efficiency. As for the comparison of results, it is similar to the one we saw in Table 8, where CCPM in many instances dominates MOGA.

Table 8. Setup and percentage trim loss obtained by CCPM and MOGA, (W = 5180)

Setup	MOGA	CCPM
Instance Fiber06 5180		
7		1.413
6	2.09	1.829
5	8.27	2.557
4		3.365
3		5.632
Instance Fiber07 5180		
10		1.263
9		1.784
8		2.152
7		2.836
6		3.062
5		3.475
4	4.98	4.264
3	8.16	6.113
Instance Fiber08 5180		
10		2.045
9		2.468
8		2.731
7		4.012
6		4.472
5	3.26	5.469
4	4.46	7.582
3	6.86	9.381
2		10.563
Instance Fiber09 5180		
9		4.231
8		7.756
7	9.36	9.067
6		9.978
5	11.28	10.043
4		10.648
3		11.054
2		11.672
Instance Fiber10 5180		
14		1.543
13		1.879
12		2.432
11		2.874
10		3.061
9		3.453
8		3.976
7		4.120
6	4.20	4.206
5	7.17	7.132
4	47.31	10.056
3		15.216
2		20.542
Instance Fiber14 5180		
11		1.054
10		1.768
9		3.546
8		4.672
7		6.453
6	5.45	6.872
5		8.590
4		9.441
3		11.632
Instance Fiber11 5180		
12		1.575
11		2.123
10		3.857
9		4.379
8		4.937
7	2.98	5.075
6	9.13	6.654
5	12.20	7.475
4	14.67	8.536
Instance Fiber13a 5180		
14		1.854
13		2.631
12		2.753
11		3.214
10		3.879
9	4.24	5.625
8	6.06	7.541
7	10.05	10.638
6		11.057
5		12.631
4		13.542
3		16.564
2		21.076
Instance Fiber13b 5180		
9		1.112
8		1.921
7		2.027
6	2.92	2.671
5	10.26	3.758
4		6.531
3		8.042
2		11.463
Instance Fiber18 5180		
14	2.97	2.113
13	4.03	3.734
12	47.56	5.316
11		7.472
10		8.521
9		10.346
8		10.965
7		11.678
6		14.054
5		14.732
4		17.125
3		17.543
Instance Fiber20 5180		
16	3.96	2.075
15	7.11	3.326
14	32.31	5.647
13	99.81	5.874
12		6.366
11		8.529
10		10.321
9		11.654
8		13.426
7		16.563
5		17.454
Instance Fiber19 5180		
25		0.879
24		1.065
23		1.211
22		1.645
21		1.936
20		2.043
19		2.227
18		2.566
17		2.876
16		3.210
15		3.346
14		3.951
13		4.323
12		4.774
11		5.120
10	5.77	6.431
9	8.92	9.057
8		10.546
7		13.472
6		22.551
5		29.475
4		36.328
3		41.547
Instance Fiber28b 5180		
22	2.86	1.751
21	4.49	3.094
20	7.76	3.762
19	28.90	4.105
18		4.845
17		5.312
16		5.982
15		6.836
14		7.128
13		7.471
12		8.326
11		8.852
10		9.120
9		10.541
8		11.043
7		11.453
6		13.683
5		14.543
Instance Fiber17 5180		
10	3.01	2.322
9	5.30	4.231
8	9.63	8.766
7	44.62	9.775
6		10.111
5		13.012
4		15.636
Instance Fiber23 5180		
16	2.83	1.021
15	4.26	2.349
14		2.744
13		4.272
12		5.763
11		7.264

2	15.211	4	19.677	10	8.063
				9	10.211
				8	10.782
				7	12.134
				6	14.652
				5	16.212
				4	19.580

Table 9. Setup and percentage trim loss obtained by CCPM and MOGA, (W = 9080))

Setup	MOGA	CCPM
Instance Fiber07 9080		
4	5.95	1.763
3		4.262
2	11.52	5.647
1		7.551
Instance Fiber08 9080		
4	1.03	1.121
3	3.13	2.054
2	19.97	5.533
1		10.324
Instance Fiber09 9080		
5	2.86	1.768
4		2.976
3		4.421
2		5.043
1		7.665
Instance Fiber10 9080		
5	1.76	1.421
4	12.20	2.443
3		3.775
2		5.543
1		8.654
Instance Fiber11 9080		
5	2.38	1.057
4	5.07	2.648
3	11.90	2.975
2		4.073
1		7.751
Instance Fiber13a 9080		
6	2.25	1.843
5	44.25	3.612
4		4.221
3		5.475
2		7.452
1		10.761
Instance Fiber13b 9080		
5	3.08	2.371
4	28.85	3.542
3		7.683
2		12.631
		38.654
Instance Fiber14b 9080		
7	5.62	1.852
6	16.94	3.237
5		3.875
4		4.534
3		6.352
2		15.471

Setup	MOGA	CCPM
Instance Fiber15, 9080		
6	1.31	1.765
5	7.64	3.878
4	13.97	4.555
3		6.653
2		8.677
1		10.543
Instance Fiber16, 9080		
13	2.95	1.574
12	7.24	2.745
11	93.04	3.231
10		5.162
9		5.874
8		6.213
7		7.532
6		9.041
5		9. 543
4		11.112
3		15.653
2		20.434
Instance Fiber17, 9080		
12		2.041
11	3.18	2.767
10		4.422
9	16.07	6.321
8	71.96	9.032
7		9.544
6		11.999
5		12.323
4		14.655
3		17.543
2		20.765
Instance Fiber18, 9080		
9	2.34	1.058
8	4.20	2.673
7	6.06	4.012
6	6.86	4.831
5		5.063
4		6.561
3		7.842

Setup	MOGA	CCPM
Instance Fiber28b, 9080		
19	2.42	2.202
18	5.23	3.743
17	23.50	5.372
16	32.54	6.865
15		7.428
14		7.881
13		8.455
12		10.463
11		11.054
10		11.658
9		13.124
8		13.461
7		15.547
6		16.934
5		18.725
Instance Fiber26, 9080		
15	1.00	1.074
14		1.762
13	1.94	2.875
12	17.89	3.764
11		5.547
10		9.056
9		10.423
8		13.522
7		14.053
6		14.346
5		16.871
4		18.346
Instance Fiber29, 9080		
13	3.03	2.054
12		3.021
11	5.89	3.213
10		4.890
9		6.743
8		7.047
7		7.852
6		9.362
5		10.541
4		11.752
3		16.445
2		25.322

4.2. Discussions

The algorithm proposed in this study have been tested on several real samples, either in instances, come from a chemical fiber company in Japan, or in film industry in different sizes using the Delphi program on a computer with the specifications following: Xeon® Silver 4110 machine (2.1 GHz/8 cores/11.00 MB/85 W) Ram 16 GB, DDR4 2400T HDD 1 TB Sata DVD-RW Windows 10 Pro 64 bit. It has proven its effectiveness especially for the problems studied in this work where the execution time in most cases did not exceed one minute.

In this pilot study, we noted that CCPM is a competitive method because it provides a set of solutions that are generally better than single-criteria methods. Moreover, it often dominates compared to the bi-objective method MOGA.

5. Conclusions and future work

The work presented in this article is based on the use of the multi-objective approach to solve a one-dimensional cutting stock problem with setup cost; the problem therefore consists in minimizing two objective functions: the total trim loss of the material first and the number of setups. Under duress to satisfy the demand, this approach is different to the single-objective combinatorial optimization approach in different ways, for example the notion of an optimal solution resulting from a single evaluation of several costs of an objective function may not always be the best way to solve the problem especially if it is possible to evaluate each cost separately, as in the case of cutting stock problems when it comes to minimizing the total trim loss and the number of setups. Where the independent evaluation of each objective makes it possible to calculate a set of compromise solutions for which there is no other solution that is better on each of the objectives. As these solutions are called efficient solutions, this would make the study more realistic. For this we have presented a new method based on the generation of cutting patterns and the constriction of cutting planes to seek efficient solutions to this extension of the cutting stock problem. The experiments show that the proposed method often gives better results compared to the methods of the literature. At the end of this work, we emphasize that one future course of action is to adapt the proposed approach to the two-dimensional cutting stock problem. Additionally, supporting two-step solutions can improve our approach and increase its efficiency.

References

Aliano Filho, A., Moretti, A. C., Pato, M. V. & Oliveira, W. A. (2017). A comparative study of exact methods for the biobjective integer one-dimensional cutting stock problem, *Journal of the Operational Research*, 69(1), 91–107.

Arbel, A. (1993). Large-scale optimization methods applied to the cutting stock problem of irregular shapes. *The International Journal of Production Research*, 31(2), 483–500. doi:10.1080/00207549308956738.

Araujo, S., Poldi, K. C. & Smith. J. (2014). A genetic algorithm for the one-dimensional cutting stock problem with Setups. *Pesquisa Operacional*, 34(2): 165-187. doi:10.1590/0101-7438. 2014.034.020165.

Campello, S. C., Ghidini, C. T., Ayres, A. O. C. & Oliveira. W. A. (2020). A multi- objective integrated model for lot sizing and cutting stock problems, *Journal of the Operational Research Society*, 71: 9, 1466-1478.

Cui, Y. & Yang, Y. (2010). A heuristic for the one-dimensional cutting stock problem with usable leftover. *European Journal of Operational Research*, 204(2), 245–250.

Cui, Y., Zhong, C. & Yao, Y (2015). Pattern-set generation algorithm for the one-dimensional cutting stock problem with setup cost, *European Journal of Operational Research*, 243(2). doi: 10.1016/j.ejor.2014.12.01.

Diegel, A., Montocchio, E., Walters, E., Schalkwyk, S. & Naidoo. S. (1996). Setup minimizing condition in the trim loss problem. *European Journal of Operational Research*, 95, 631-640.23, 483-493.

Delorme, M., Iori, M. & Martello, S., (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255 (1), 1 – 20.

Ehrgott, M. & Gandibleux, X. (2000). A survey and annotated bibliography of multi-objective combinatorial optimization. *Operational Research Spektrum*, 22, 425-460.

Farley, A. A. (1988). Mathematical programming models for cutting-stock problems in the clothing industry. *Journal of the Operational Research Society*, 39(1), 41–53.

Filho, A. A., Moretti, A. C. & Pato, M. V. (2018). A comparative study of exact methods for the bi-objective integer one-dimensional cutting stock problem. *Journal of the Operational Research Society*, 69(1), 91–107.

Gilmore, P. C. & Gomory, R. E. (1961). A Linear programming approach to the cutting stock problem. *Operations Research*, 9, 849-859.

Gilmore, P. C. & Gomory, R. E. (1963). A linear programming approach to the cutting stock problem Part II. *Operations Research*, 11 (6), 863–888.

Gilmore, P. C. & Gomory, R .E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations Research.*, 13 (1), 94–120.

Golfeto, R., Moretti, A. C. & Salles-Neto. L. L. (2009b). A genetic symbiotic algorithm applied to the cutting- stock problem with multiple objectives. *Advanced, Modeling and Optimization*, 11, 473-501.

Haessler, R. W. (1975). Controlling cutting pattern changes in one-dimensional trim Problems, *Operations Research*. 23, 483-493.

Haessler, R. W. (1980). A Note on computational modifications to the Gilmore –Gomory cutting stock problem. *Operations Research*, 28, 1001-1005.

Kolen, A. W. J. & Spieksma, F. C. R. (2000). Solving a bi-criterion cutting stock problem with openended demand: a case study. *Journal of The Operational Research Society*, 51, 1238-1247.

Lee, J. (2007). In situ column generation for a cutting-stock problem. *Computers & Operations Research*, 34, Issue 8, 2345-2358.

Martinovic, J., Scheithauer, G. & Valério de Carvalho, J. M. (2018). A comparative study of the arc flow model and the one-cut model for one-dimensional cutting stock problems, *European Journal of Operational Research*, 266 (2), 458-471.

Pazand, K. & Mohammadi. A. (2009). Extended haessler heuristic algorithm for cutting stock problem: a case study in film Industry. *Australian Journal of Basic and Applied Sciences*, 3(4): 3944-3953, ISSN 1991-8178.

Salles-Neto, L. L., Rufiàn-Lizana, A. M., Arana-Jimenez & Ruiz-Garzon. G. (2010). Weak efficiency in Cutting-stock problem, *Congresso nacional de matemática aplicada e computational*, 33. v. 3, p. 652-658. ISSN 1984-820X.

Suliman, S. M. (2001). Pattern generating procedure for the cutting stock problem. *International Journal of Production Economics*, 74(1-3), 293–301. doi :10.1016/S0925-5273(01)00134-7.

Tanir, D., Ugurlu, O., Guler, A. & Nuriyev, U. (2019). One-dimensional cutting stock problem with divisible items: a case study in steel industry, *TWMS Journal of Applied and Engineering Mathematics*, V.9, N.3, 2019. 473- 484.

Umetani, S., Yagiura, M. & Ibaraki, T. (2003). An LP-based local search to the one-dimensional cutting stock problem using a given number of cutting patterns. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E86- A, 1093–1102.

Wäscher, G. & Henn, S. (2013). Extensions of cutting problems, setups. *Pesquisa Operational*, Vol 33 (2). 133-162.