Fuzzy Constructive Heuristics

José A. Moreno Pérez and J. Marcos Moreno Vega

Dpto. de E.I.O. y Computación
Centro Superior de Informática
Universidad de La Laguna
38271 La Laguna, Santa Cruz de Tenerife, Spain
jamoreno@ull.es, jmmoreno@ull.es

Abstract. We consider the design of fuzzy constructive heuristic algorithms for solving combinatorial problems. The fuzzy technology is used in constructive heuristics to select the item to be included in the solution and to stop the search. This is done by considering fuzzy sets of promising elements and satisfactory solutions. The procedures that implement several of these algorithms were tested on moderate and large size instances of a well known cutting problem. These procedures show better performance than the best known heuristic for them.

1 Introduction

In a constructive method an element is iteratively added to an initially empty structure until a solution of the problem is obtained. The choice of the item to be included in the partial solution is based on one or several heuristic evaluations that measure the convenience of considering the item as belonging to the solution. The heuristic functions depend on the problem and also on the knowledge of the decision maker about the problem.

The heuristic functions are used to intelligently guide the process of searching for solutions with high quality. If the evaluation of an element depends on the items already in the solution, the function and the method are adaptive. In addition to the heuristic function, it is necessary a strategy to select the elements. One of the most known strategy is the *greedy* rule, which selects the element that optimizes the heuristic function. However, this strategy shows poor performance in most cases.

In some cases, after knowing the evaluation of an element, the expert provides comments like: *high recommendable element*, *quite good item*, *acceptable enough element*, etc. In those cases, it is possible to construct fuzzy sets of elements from which to choose the element to be included in the solution.

One of the most relevant questions, with a high effect on the quality of the solutions, is the stopping criteria applied. As indicated above, the usual expressions are: *acceptable solution*, *good enough solution*, etc. The construction of a fuzzy set from the desirable situation, allows us to design appropriate stopping criteria.

In this paper we introduce the fuzzy constructive methods and design fuzzy stopping rules. Moreover, we show the performance of the proposed method on a well known cut problem.

2 Fuzzy Constructive Methods

Let $E = \{e_1, e_2, ..., e_n\}$ be the set of elements to be included in the solution and f an adaptive heuristic function. The functions that evaluate the elements are such that better elements have smaller values. Let X denote the structure that stores the obtained partial solution so far, and then $T = E - X$ is the set of elements not included in the partial solution.

The pseudocode of a classical constructive method with the greedy strategy is shown in figure 1. The main drawback of the greedy strategy is that, in general, it provides low quality solutions. It is true even if an improvement procedure (a local search) is applied to the obtained solution.

```
Procedure Greedy_Constructive_Method
    begin
        $X = \emptyset$;
        $T = E$;
        repeat
                Evaluate($f(e) : e \in T$);
                $e^* = \arg\min_{e \in T}\{f(e)\}$;
                $X = X \cup \{e^*\}$;
                $T = T - \{e^*\}$;
        until ($X$ is feasible);
        return $X$;
    end.
```

Fig. 1. Greedy Constructive Method

Let us assume that there is a membership function $\mu(.)$ that evaluates the degree of belonging of an element $e \in E$ to the set of *best elements*. The membership function $\mu(e)$ of an element e depends on its heuristic evaluation, $f(e)$. Then the set of *best elements* E^* of E is constructed by:

$$E^* = \{e \in E / \mu(e) \geq \alpha\},$$

i.e., E^* is the α-cut of μ ($\alpha \in [0, 1]$ is fixed by the decision maker).

Therefore, the pseudocode of the fuzzy constructive method that we propose is as shown in figure 2.

In this pseudocode, by $Choose(e \in E^*)$ we mean the method used to determine the element $e \in E^*$ that is included in the partial solution X.

```
Procedure Fuzzy_Constructive_Method
    begin
      $X = \emptyset$;
      $T = E$;
      repeat
            Evaluate($f(e) : e \in T$);
            Generate($E^*$);
            Choose($e^* : e^* \in E$);
            $X = X \cup \{e^*\}$;
            $T = T - \{e^*\}$;
      until ($X$ is feasible);
      return $X$;
    end.
```

Fig. 2. Fuzzy Constructive Method

Some alternatives for this method are the equiprobable sampling in E^* or the proportional sampling in E^*. They are:

Equiprobable sample:

$$\Pr(\textit{choose } e) = \frac{1}{|E^*|}.$$

Proportional sample:

$$\Pr(\textit{choose } e) = \frac{\mu(e)}{\sum_{e \in E^*} \mu(e)}.$$

Note that, in contrast to the classical constructive method, several executions of the fuzzy constructive method will provide different solutions. In that way, the probability of getting high quality solutions increases.

Figure 3 shows the iterated fuzzy constructive method. It includes an improvement method after each execution of the fuzzy constructive procedure. In this pseudocode, $Improving\_Method(X, X')$ returns an improved solution X' obtained from X.

The improving method applied in this procedure can be, for example, Fuzzy Adaptive Neighbourhood Search (FANS) [7], but can also be a crisp procedure like any local search.

3 Fuzzy Stopping Rules

Ideally, the search process must finish when the optimal solution of the problem is found. However, this stopping criterion is not applicable in real situations, since the optimal solution is unknown. Therefore, a real stopping criterion is to finish the search when a high quality solution that satisfies the preference of the decision maker is met.

```
Procedure Iterated_Fuzzy_Constructive_Method
    begin
        Fuzzy_Constructive_Method;
        Improving_Method($X$, $X'$);
        $X^* = X'$;
        repeat
                Fuzzy_Constructive_Method;
                Improving_Method($X$, $X'$);
                if ($X'$ is better than $X^*$) then
                    $X^* = X'$;
        until (stopping_rule);
        return($X^*$);
    end.
```

Fig. 3. Iterated Fuzzy Constructive Method

For many problems, the quality of the solution can be measured by some characteristics. These characteristics indicate that the reached situation is *acceptable*, *good enough*, *difficult to improve*, ... These situations are determined by a set of variables that characterize the quality of the solution. Using these variables, some fuzzy sets of solutions are used to provide appropriated fuzzy stopping criteria. The criteria consist of stopping the search when a solution of these fuzzy set is met.

4 The non-guillotine rectangular two-dimensional cutting problem

The two-dimensional cutting problem consists of determining the optimal cut of a single sheet into a set of small pieces of given shapes and sizes. In the rectangular two-dimensional cutting problems, the stock sheet and the small pieces have rectangular shape. The optimality of the cut is given by the amount of space used or the waste material. In other problems, given the number of small pieces with economical values, the solution consists of selecting pieces to cut in order to maximize the total value. These problems appear in relevant commercial and industrial application areas, where one or several big sheets of wood, cloth, paper or metal have to be cut in a large number of small pieces (see, for example, [1] and [3]).

The non-guillotine rectangular two-dimensional cutting problem [4], [5] is formulated as follows. Given a rectangular sheet of a fixed width w and unlimited high, and a set of small rectangles

$$\mathcal{R} = \{R(w_1, h_1), R(w_2, h_2), \ldots, R(w_n, h_n)\},$$

given by the lengths of their sides, determine the way to cut all the pieces in a sheet with minimum total height. In this problem the cuts can be of

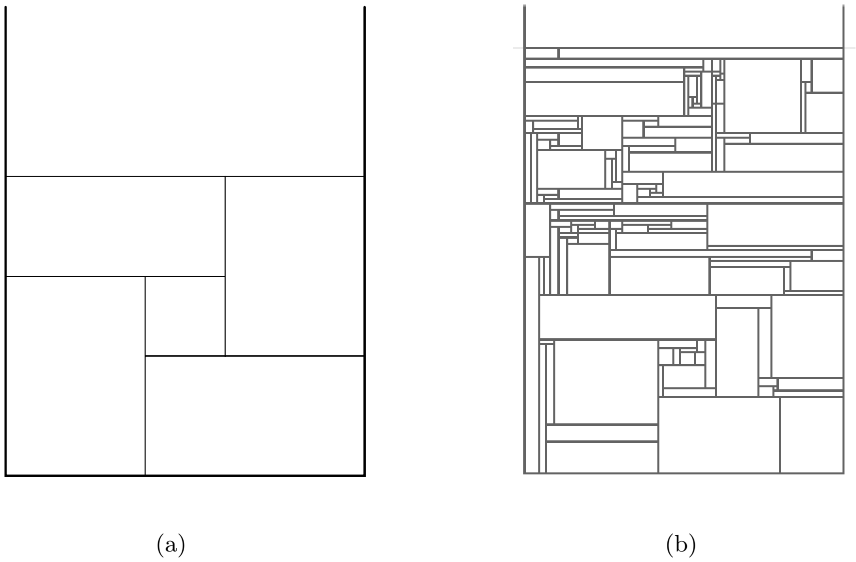

Fig. 4. Rectangular cutting problems. **(a)** Non guillotine cutting. **(b)** Non-guillotine rectangular two-dimensional cutting problem

non-guillotine type and the objects can be rotated. The guillotine cuttings are those that go from one edge of the stock rectangular sheet to the opposite edge. In a non guillotine cutting, this is not true (see figure 4(a)). Obviously, each small piece has to be with one of its sides parallel to the fixed edge of the stock sheet in the optimal cut. Thus, the values w_i and h_i are named the width and high of the pieces and can be rotated 90^0. At least one of the two sides, w_i or h_i, of each rectangle is assumed to be smaller than the width w.

4.1 Contour

A partial solution is given by the position of the bottom-left corner of each small rectangle (and the boolean variable to know if the piece is rotated or not). The solution is feasible if there is not an overlap between two rectangles. The rectangles are obviously shifted down in order to minimize the total height wasted. The partial solution determines the waste of space determined by the fixed width side, the unlimited edges and an upper piecewise rectilinear contour as shown in figure 5. In addition, not useful areas called trim loss can be generated (see the partial solution obtained after the insertion of the rectangle 4 in figure 5).

Let us assume that $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$, where $\mathcal{R}_1$ is the set of rectangles already packed in the partial solution and $\mathcal{R}_2 = \mathcal{R} \setminus \mathcal{R}_1$ the set of remainder rectangles to be packed.

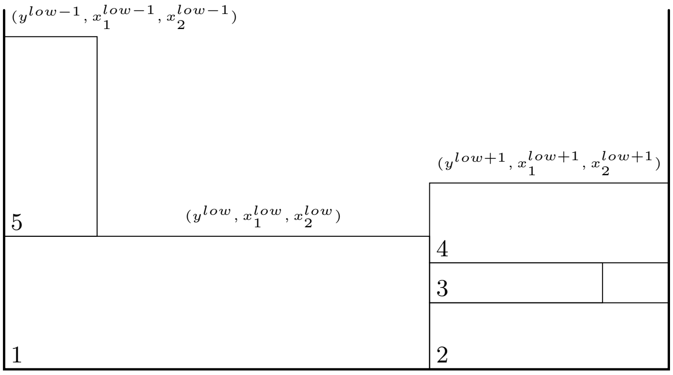
$(y^{low-1}, x_1^{low-1}, x_2^{low-1})$
$(y^{low+1}, x_1^{low+1}, x_2^{low+1})$
$(y^{low}, x_1^{low}, x_2^{low})$
5
4
3
1
2

Fig. 5. Contour

Consider the upper envelope of the set of rectangles already included in the partial solution. This envelope is given by a sequence of segments, $\mathcal{C} = (s^1, s^2, ..., s^c)$, where each segment s^i is given by the height and the position of the extremes, i.e., $s^i = (y^i, x_1^i, x_2^i)$. The contour $\mathcal{C}$ verifies $x_1^1 = 0$, $x_2^c = w$ and, for $1 \leq i < c - 1$, $x_2^i = x_1^{i+1}$. Then,

$$\mathcal{C} = \{(y^1, x_1^1, x_2^1), (y^2, x_1^2, x_2^2), ..., (y^c, x_1^c, x_2^c)\}.$$

Let $s^{low} = (y^{low}, x_1^{low}, x_2^{low})$ be the segment with small height; i.e. such that $y^{low} = \min_{i=1,...,c}\{y^i\}$.

Let us assume that the width of any not packed rectangle is the side closer to the size of the lowest segment $w^{low} = |x_2^{low} - x_1^{low}|$ of the upper contour. We can do this assumption since otherwise a simple rotation of the object will provide this hypothesis.

Since we want to obtain the packing with smallest height, we can design several natural constructive ways in which the rectangles are packed in the segment with smallest height C. Among all the possible alternatives, we describe three of them that also take into account other partial objectives.

4.2 Sets of best rectangles

Consider the situation shown in figure 5 where the lowest segment is $s^{low} = (y^{low}, x_1^{low}, x_2^{low})$. The next and before segments are

$$s^{low+1} = (y^{low+1}, x_1^{low+1}, x_2^{low+1})$$

and

$$s^{low-1} = (y^{low-1}, x_1^{low-1}, x_2^{low-1}).$$

Let us assume, without loss of generality, that the heights of these segments are such that $y^{low} \leq y^{low+1} \leq y^{low-1}$.

From the segment $s^{low} = (y^{low}, x_1^{low}, x_2^{low})$, there are several ideal situations that could be achieved by the inclusion of a new rectangle. A very good situation appears if we can include a rectangle with width equal to s^{low}. Another very good situation appears if the height of the included rectangle is close to $y^{low+1} - y^{low}$ or to $y^{low-1} - y^{low}$ (see figure 5). Therefore, there are three notions of ideal situations that are used to build fuzzy sets of best rectangles. The membership functions of these sets are as follows.

The first fuzzy set of best rectangles corresponds to those that most fit to the width w^{low}. Let $\alpha_1 \in [0,1]$ be a fit threshold given by the decision maker. Then the membership function of any object $e = R(w,h) \in \mathcal{R}_2$ is given by:

$$\mu^1(R(w,h),\alpha_1) = \begin{cases} \frac{1}{\alpha_1}(\alpha_1 - w^{low} + w) & \text{if } w^{low} - \alpha_1 \leq w \leq w^{low} \\ \\ 0 & \text{otherwise} \end{cases}$$

This membership function defines the first fuzzy set $\mathcal{E}_1$ of the *best elements* on the universe of the not packed rectangles.

A second fuzzy set of best rectangles, $\mathcal{E}_2$, is obtained by considering the set of not packed rectangles such that, if one of them is packed on s^{low} it will provide a segment with height close to y^{low+1} (see figure 5).

Let $\alpha_2 \in [0,1]$ be a second fit threshold provided by the decision maker. Let the second membership function of an object $e = R(w,h)$ be given by:

$$\mu^2(R(w,h),\alpha_2) = \begin{cases} \frac{1}{\alpha_2}|\alpha_2 - y^{low+1} + y^{low} - h| & \text{if } |y^{low+1} - y^{low} + h| \leq \alpha_2 \\ \\ 0 & \text{otherwise} \end{cases}$$

Similarly, consider the set $\mathcal{E}_3$ of the not packed rectangles such that, if one of them is packed on s^{low} the corresponding segment has height very close to y^{low-1}.

Let $\alpha_3 \in [0,1]$ be a third fit threshold provided by the decision maker. Let a third membership function of an object $e = R(w,h)$ be given by:

$$\mu^3(R(w,h),\alpha_3) = \begin{cases} \frac{1}{\alpha_3}|\alpha_3 - y^{low-1} + y^{low} - h| & \text{if } |y^{low-1} - y^{low} - h| \leq \alpha_3 \\ \\ 0 & \text{otherwise} \end{cases}$$

We have three rules for selecting the new rectangle to be included in the partial solution. These rules consist in selecting a rectangle from $\mathcal{E}_1$, from $\mathcal{E}_1 \cap \mathcal{E}_2$ and from $\mathcal{E}_1 \cap \mathcal{E}_3$. With the first rule, from the list

$$\mathcal{L}_1 = \{R(w,h) \in \mathcal{R}_2 : \mu^1(R(w,h),\alpha_1) \geq 0\},$$

an element is chosen to be packed in the partial solution. For the second rule, from the list

$$\mathcal{L}_2 = \{R(w,h) \in \mathcal{L}_1 : \mu^2(R(w,h),\alpha_2) \geq 0\},$$

an element is chosen to be packed in the partial solution. Finally, for the third rule, from the list

$$\mathcal{L}_3 = \{R(w,h) \in \mathcal{L}_1 : \mu^3(R(w,h),\alpha_3) \geq 0\},$$

an element is chosen to be packed in the partial solution.

In any case, if the corresponding list $\mathcal{L}_i (i = 1, 2, 3)$ is empty, we take from $\mathcal{R}_2$ the rectangle that better fits to s^{low}. If such rectangle does not exist, we rebuild the contour C replacing the segments s^{low} and s^{low+1} by a new segment $s = (y^{low+1}, x_1^{low}, x_2^{low+1})$. The area between x_1^{low}, x_2^{low}, y^{low} and y^{low+1} is wasted.

4.3 Fuzzy stopping rules for the cutting problem

Besides the objective value, there are other values that can be used to evaluate the quality of a solution for the non-guillotine rectangular two-dimensional cutting problem. Some of them are the total area of trim loss and the shape of superior contour.

Ideally, we want to pack the rectangles in such a way that the total area of trim loss is zero and the upper contour is smooth enough.

Given a solution X with objective value $f(X)$, let $TrimLoss(X)$ be the total area of trim loss of X and $Shape(X)$ be the average distance of the upper contour to the total height of X. This average distance is computed by:

$$Shape(X) = \frac{1}{c}\sum_{i=1}^{c} |f(X) - y^i|.$$

Let $\beta_1 \in [0,1]$ a value fixed by the decision maker. The set of solutions with an acceptable trim loss is given by the following membership function μ^4. Let X be a solution, then:

$$\mu^4(X,\beta_1) = \begin{cases} \frac{1}{\beta_1}(\beta_1 - TrimLoss(X)) & \text{if } 0 \leq TrimLoss(X) \leq \beta_1 \\ \\ 0 & \text{otherwise} \end{cases}$$

Let

$$\mathcal{S}_1 = \{X : \mu^4(X,\beta_1) \geq 0\}.$$

Similarly, let $\beta_2 \in [0,1]$ be another value fixed by the decision maker. The set of solutions with a smooth enough contour is given by the membership function μ^5. This membership function for a solution X is given by:

$$\mu^5(X,\beta_2) = \begin{cases} \frac{1}{\beta_2}(\beta_2 - Shape(X)) & \text{if } 0 \leq Shape(X) \leq \beta_2 \\ \\ 0 & \text{otherwise} \end{cases}$$

Let

$$\mathcal{S}_2 = \{X : \mu^5(X,\beta_2) \geq 0\}.$$

The proposed stopping criterion combines the above sets of solutions. The search stops when a solution of $\mathcal{S}_1 \cap \mathcal{S}_2$ is reached.

5 Computational experiments

The computational experiments were performed in two phases. In the first phase, the values of the parameters that define the fitness of the rectangles were determined. Then, the proposed constructive methods were compared with the best known procedure for solving the problem. In a second phase, the performance of the three constructive methods to solve moderate and large size instances of the problem was analyzed.

5.1 Tuning the parameters

We have three constructive methods obtained by using the lists of best rectangles $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$. Let FCM_1, FCM_2 y FCM_3 denote these methods. The stopping rule used (see figure 3) was to fix the number of iterations of the loop to $n_{iter} = 40$.

To fix the parameters (α_i, $i = 1,2,3$) that control the fitness of the rectangles to the lowest segment, randomly generated packing problems were solved. Several values for α_i, $i = 1,2,3$ where fixed and each method was run 5 times for the instances generated. The output variable was the average objective value in the n_{iter} executions of the loop.

For each parameter three levels were taken: $\alpha_1 = 0, 0.1, 0.2$, $\alpha_2 = 0, 0.1, 0.2$ and $\alpha_3 = 0, 0.1, 0.2$ (with $\alpha_2 = \alpha_3$). Therefore we considered the treatments T_{α_1}, T_{α_1,α_2} and $T_{\alpha_1,\alpha_2,\alpha_3}$.

We applied the Friedman nonparametric test to analyze the data (see [2]), since the previous normality and variance equality tests were negative. When the null hypothesis of equality between treatment was rejected, we applied the Friedman multiple comparison tests ([2], page 274) to obtain the significative difference between treatments.

In tables 1 and 2 we show the p-value associated with the Friedman statistic for the problem. In addition, for those problems where the null hypothesis of the equality between treatments was rejected, we show the treatment with the smallest average value and, when the test did not give significative differences, we also show the treatment with the second smallest average value (between round brackets).

From the results of the tests we conclude:

1. *FCM_1:* the appropriated value for α_1 is 0; i.e., that corresponding to the best fit;
2. *FCM_2:* the values of the parameters depend on the size of the problem. For problems with 200 or more rectangles, the treatment that gives the best performance is $T_{0,0}$, with the only exception $n = 200$, $w = 60$ and $h_{opt} = 100$. For problems with 100 rectangles, there is not a clear conclusion, since it could be chosen between the treatments $T_{0,0}$ and $T_{0,0.2}$. We selected $T_{0,0.2}$ taking into account the number of times that both $T_{0,0}$ and $T_{0,0.2}$ appear as the best or second best treatment. For $n = 50$, the ties is even more clear if we compare the number of times that a treatment appears as the best one. Then, taking into account the second best treatment, we selected $T_{0,0.1}$.
3. *FCM_3:* using the above arguments it follows that, for problems with 200 or less rectangles, the recommended choices are $\alpha_1 = 0$ and $\alpha_2 = \alpha_3 = 0.2$. For problems with 300 rectangles, the choice is $\alpha_1 = \alpha_2 = \alpha_3 = 0$.

5.2 Comparative

To compare the efficiency and the efficacy of our proposals with respect to Simulated Annealing that uses the Bottom-Left ($SA+BL$) strategy (see [4]), we used the test bed described in [4] (the data of these instances can be found in [6]). It is a bed of 21 instances arranged in 7 categories of 3 instances.

In the first four columns of table 3 we show the characteristics of these problems. The fifth column shows the best objective value and the computational time (in minutes) needed by $SA + LB$. The values have been taken from [4]. From columns 6 to 8 we show the best objective values obtained with FCM_1, FCM_2 and FCM_3. For these methods we do not show the computational time because it was insignificant.

The first notable improvement in the performance is shown by the considerable decreasing in the running time when we use any of our proposals. For instance, to the category C_7 it goes from 4181 minutes to a insignificant time. In addition, the efficacy of FCM_1, FCM_2 and FCM_3 is comparable or better than that of $SA + BL$. Among the three proposals, the best performance is, in general terms, FCM_3. This behavior can also be seen in the results obtained for problems with bigger sizes (see table 4).

5.3 Results for large instances

Table 4 shows the results obtained on random instances with large sizes. The three first columns show the number of rectangles, the width of the sheet and

Table 1. Significance levels (p-value) for the equality among treatments and best treatments (between brackets, second best one)

n	w	h_{opt}		FCM_1	FCM_2	FCM_3
50	30	45	p-value	0.008	0.000	0.000
			treatment	T_0	$T_{0,0.1}$	$T_{0,0}$ ($T_{0,0.2}$)
		60	p-value	0.022	0.005	0.244
			treatment	T_0	$T_{0.1,0.2}$ ($T_{0,0.1}$)	=
	60	90	p-value	0.022	0.026	0.001
			treatment	T_0	$T_{0.1,0.2}$ ($T_{0,0}$)	$T_{0.1,0}$ ($T_{0,0.2}$)
		100	p-value	0.819	0.000	0.000
			treatment	=	$T_{0,0.2}$ ($T_{0,0.1}$)	$T_{0.1,0}$ ($T_{0,0.2}$)
	90	120	p-value	0.007	0.002	0.010
			treatment	T_0	$T_{0.1,0}$ ($T_{0,0}$)	$T_{0,0.2}$
		150	p-value	0.022	0.001	0.156
			treatment	T_0	$T_{0,0.1}$	=
100	50	70	p-value	0.007	0.001	0.000
			treatment	T_0	$T_{0,0}$ ($T_{0,0.2}$)	$T_{0,0}$ ($T_{0,0.2}$)
		110	p-value	0.007	0.000	0.000
			treatment	T_0	$T_{0,0.2}$	$T_{0,0.2}$ ($T_{0,0}$)
	80	90	p-value	0.022	0.003	0.032
			treatment	T_0	$T_{0,0.1}$ ($T_{0,0.2}$)	$T_{0,0}$ ($T_{0,0.2}$)
		140	p-value	0.007	0.001	0.009
			treatment	T_0	$T_{0,0.2}$	$T_{0,0.1}$ ($T_{0,0.2}$)
	100	180	p-value	0.022	0.003	0.000
			treatment	T_0	$T_{0,0}$	$T_{0,0}$ ($T_{0,0.2}$)
		200	p-value	0.247	0.000	0.000
			treatment	=	$T_{0,0}$	$T_{0.1,0}$ ($T_{0,0.2}$)

the optimal objective value. The last three columns show the best average objective value in 5 runs of FCM_1, FCM_2 and FCM_3. From these results we conclude that FCM_3 is slightly better than FCM_1 and FCM_2, and the good performance of FCM_1, FCM_2 and FCM_3. As indicated above, the time required is not significant.

5.4 Stopping rule

To evaluate the quality of the proposed stopping rule, randomly generated instances of the cutting problem were solved using FCM_3. Each instance was solved 5 times. Table 5 shows the best average objective values and the average number of iterations for those runs. Three values for the parameter β_1 and two values for β_2 were fixed. For β_1 we use $0.05A$, $0.03A$ and $0.02A$ where A is the total area of the rectangles

$$A = \sum_{i=1}^{n} w_i h_i .$$

Table 2. Significance levels (p-value) for the equality among treatments and best treatments (between brackets, second best one)

n	w	h_{opt}		FCM_1	FCM_2	FCM_3
200	60	100	p-value	0.074	0.004	0.001
			treatment	T_0	$T_{0.2,0.2}$	$T_{0.1,0.2}$ ($T_{0.2,0.2}$)
		130	p-value	0.007	0.000	0.000
			treatment	T_0	$T_{0,0.2}$ ($T_{0,0}$)	$T_{0,0}$ ($T_{0,0.2}$)
	100	90	p-value	0.007	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0.1,0.2}$ ($T_{0,0.2}$)
		140	p-value	0.022	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0,0.1}$ ($T_{0,0.2}$)
	200	150	p-value	0.015	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0.1,0}$ ($T_{0,0.2}$)
		200	p-value	0.022	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0,0}$ ($T_{0,0.2}$)
300	80	150	p-value	0.022	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0,0}$
		200	p-value	0.007	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0,0}$
	140	120	p-value	0.022	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0,0}$
		200	p-value	0.022	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0,0.1}$ ($T_{0,0}$)
	200	200	p-value	0.007	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0,0}$
		280	p-value	0.022	0.000	0.000
			treatment	T_0	$T_{0,0}$	$T_{0,0}$

The two values for β_2 are $\beta_2 = 3$ and $\beta_2 = 1.5$. From the results obtained we conclude that:

1. For a fixed value of β_1, better solutions were obtained by decreasing β_2.
2. The best objective values were obtained for small values of the parameters β_1 y β_2.
3. The number of required iterations is small enough. This indicates that the constructive method FCM_3 provides solutions of high quality.

References

1. Bischoff, E.E., Wäscher, G. (1995) Cutting and Packing. European Journal of Operational Research **84** , 503-505
2. Daniel, W.W. (1990) Applied Nonparametric Statistics. PWS-Kent Publishing Company, Boston
3. Dyckhoff, H. (1990) A Typology of Cutting and Packing Problems. European Journal of Operational Research **44** , 145-159

Table 3. Best objective values and required times in minutes (average values for category)

Category	n	w	h_{opt}		$SA+BLF$	FCM_1	FCM_2	FCM_3
C_1	16 or 17	20	20	objective	20.8	22.6	22	22
				time	0.7			
C_2	25	40	15	objective	15.9	17	17	16.33
				time	2.4			
C_3	28 or 29	60	30	objective	31.5	33.66	35.33	33.66
				time	4			
C_4	49	60	60	objective	61.8	62.66	64.33	63
				time	33			
C_5	72 or 73	60	90	objective	92.7	94.33	94	93
				time	115			
C_6	97	80	120	objective	123.6	125.33	124.33	124
				time	382			
C_7	196 or 197	160	240	objective	249.6	247	245	246
				time	4181			

Table 4. Random instances: average objective values

n	w	h_{opt}	FCM_1	FCM_2	FCM_3
50	50	50	52	52	51.8
50	40	60	62.6	63.4	62
100	50	50	52	51.4	51.2
100	50	75	76.8	77	77
200	100	100	101.4	102	101.8
200	120	160	163	162.4	163
500	100	200	202.2	202.8	202.6
500	150	200	202.2	202	202
700	250	320	322.6	323	322.6
700	250	400	403.2	403.6	403

Table 5. Stopping rule: average objective values and iterations

				$\beta_1 = 0.05$		$\beta_1 = 0.03$		$\beta_1 = 0.02$	
				$\beta_2 = 3$	$\beta_2 = 1.5$	$\beta_2 = 3$	$\beta_2 = 1.5$	$\beta_2 = 3$	$\beta_2 = 1.5$
n	w	h_{opt}							
50	40	60	objective	63.2	62	63.4	62.2	62.6	62
			iteration	3.6	16.6	3	21.8	4	10.2
50	50	50	objective	51.8	51.8	52	51.4	52	51.6
			iteration	1.4	4.2	1.8	11.8	9.8	16.8
100	50	50	objective	52.4	51.8	51.8	52	51.6	51.8
			iteration	1	1	1	1.8	1	1.8
100	50	75	objective	77.2	77	77.4	77	77.4	76.8
			iteration	1.2	2	1	1.4	2	2
200	100	100	objective	102	102	102.2	102	102	102
			iteration	1.2	1.4	1	1.2	1	1.4
200	120	160	objective	163.4	62.4	162.8	162	162.8	162
			iteration	1.8	10.6	2.2	13.2	7.6	10.8

4. Hopper, E., Turton, B. (2001) An Empirical Investigation of Meta-heuristic and Heuristic Algorithms for a 2D Packing Problem. European Journal of Operational Research **128** , 34-57
5. Hopper, E., Turton, B. (2001) A Review of the Application of Meta-Heuristic Algorithms to 2D Strip Packing Problems. Artificial Intelligence Review **16**, 257-300
6. *OR Library.* `http://mscmga.ms.ic.ac.uk/jeb/orlib/stripinfo.html`)
7. Pelta, D., Blanco, A., Verdegay, J. L., (2002) *Fuzzy Adaptive Neighborhood Search: Examples of Applications* In this same volume