Hyperheuristic Encoding Scheme for Multi-Objective Guillotine Cutting Problems

Jesica de Armas
Dpto. Estadistica, I. O. y Computación
Universidad de La Laguna
38271, Tenerife, Spain
jdearmas@ull.es

Gara Miranda
Dpto. Estadistica, I. O. y Computación
Universidad de La Laguna
38271, Tenerife, Spain
gmiranda@ull.es

Coromoto León
Dpto. Estadistica, I. O. y Computación
Universidad de La Laguna
38271, Tenerife, Spain
cleon@ull.es

ABSTRACT

Most research on Strip Packing and Cutting Stock problems are focused on single-objective formulations of the problems. However, in this work we deal with more general and practical variants of the problems, which not only seeks to optimise the usage of the raw material, but also the overall production process. The problems target the cutting of a large rectangle in a set of smaller rectangles using orthogonal guillotine cuts. Common approaches are based in the minimisation of the strip length required to cut the whole set of demanded pieces (for strip problems) and in the maximisation of the total profit obtained from the available surface (for cutting stock problems). In this work we also deal with an extra objective which seeks to minimise the number of cuts involved in the cutting process, thus maximising the efficiency of the global production process. In order to obtain solutions to these problems, we have applied some of the most-known multi-objective evolutionary algorithms, since they have shown a promising behaviour when tackling multi-objective real-world problems. We have designed and implemented hyperheuristic-based encodings as an alternative to combine heuristics in such a way that a heuristic's strengths make up for the drawbacks of another.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Inteligence—*Problem Solving, Control Methods and Search* Heuristic Methods

General Terms

Algorithms

Keywords

Multi-Objective Optimisation, Cutting Problems, Evolutionary Algorithms, Encoding Schemes, Hyperheuristics

1. INTRODUCTION

Cutting Problems [24] arise in many production industries where large stock sheets (glass, textiles, pulp and paper, steel, etc.) must be cut into smaller pieces. Although many variants of the problem have been widely studied, here we have focused on general guillotine problems which do not introduce constraints about the number of cutting stages. Guillotine problems obtain the pieces from the sheet of raw material by using exclusively orthogonal guillotine cuts. That means that any cut must run from one side of the rectangle to the other end and be parallel to the other two edges (Figure 1). The production of non-guillotinable cuts may entail a more complex machinery operation. For this reason, and in order to encompass a wider range of industrial cutting machines, we have focused on the guillotinable formulation of the problems. The paper industry is the most common real-world application which requires quillotine cutting.

The first problem we have studied is the *Two-Dimensional Guillotine Strip Packing Problem* (2DSPP). The 2DSPP arises in many production industries where the raw materials are in the form of rolls and involves the cutting of a complete set of n demanded pieces from a large stock sheet of material using guillotine cuttings. The stock sheet has fixed width W and unlimited length L. Each rectangular piece i demanded has fixed dimensions (l_i, w_i), although they can be rotated 90 degrees, thus allowing for the dimensions (w_i, l_i) as well. In this problem the goal is to minimise the strip length required to cut the whole set of demanded pieces.

The second studied problem is the *Constrained Two-Dimensional Cutting Stock Problem* (2DCSP). The 2DCSP targets the cutting of a large rectangle S of dimensions $L \times W$ in a set of smaller rectangles using orthogonal guillotine cuts. The produced rectangles must belong to one of a given set of rectangle types $\mathcal{D} = \{T_1 \ldots T_n\}$ where the i-th type T_i has dimensions $l_i \times w_i$. Associated with each type T_i there is a profit p_i and a demand constraint b_i. Usually, the main goal is to find a feasible cutting pattern with x_i pieces of type T_i maximising the total profit.

Although in both problems the common goal consists in finding the layouts which make a best usage of the available material, in some industrial fields, the raw material is either very cheap or can be easily recycled, so that, a more important criterion for the pattern generation may be the speed at which the pieces can be obtained, thus minimising the production times and maximising the usage of the cutting equipment. The cutting process is specifically limited by the features of the machinery available but, in general, it is determined by the number of cuts involved in the packing patterns.

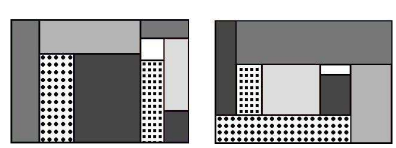

Figure 1: Guillotine and non-guillotine cuts

GECCO'11, July 12–16, 2011, Dublin, Ireland.

Moreover, the number of cuts required for the cutting process is also crucial to the life of the industrial machines. Since the number of cuts is an important aspect in determining the cost and efficiency of the production process, a comprehensive optimisation methodology should take also this criterion into consideration. Therefore, in this study, the number of cuts is taken as a second design objective in both cutting problems. This way, the problems can be posed as multi-objective optimisation problems.

Many single-objective approaches have been proposed to solve the 2DSPP. However, works dealing with the multi-objective approach are almost inexistent and they all are based on the usage of Multi-Objective Evolutionary Algorithms (MOEAs). Moreover, the approaches tackling such type of guillotine problem are based on representations where the pattern layouts are explicitly encoded using a post-fix notation. This encoding scheme is effective for the smaller test problem instances, but not for the larger [4]. For this reason, in this work, we propose a hyperheuristic encoding scheme and MOEAs to solve the larger test problem instances of the multi-objective 2DSPP. In the case of the 2DCSP, we haven't found any work dealing with a multi-objective formulation of the problem. So, taking into account the similarities with the 2DSPP, the same encoding scheme and algorithms are applied for this problem.

The remaining content of this paper is organised as follows. The state-of-the-art on the solution of the widely studied single-objective and multi-objective formulation of the problems is given in section 2. Section 3 gives a general overview of multi-objective optimisation. In section 4, we present the approaches designed to deal with the here studied multi-objective guillotine cutting problems. The experimental results of these approaches are presented in section 5. Finally, the conclusions and some lines of future work are given in section 6.

2. APPROACHES FOR 2DSPP AND 2DCSP

As occurs with most cutting and packing problems, the problem of finding an optimal solution for the 2DSPP is NP-hard, so research has focused mainly on developing heuristics which can provide good (though not necessarily optimal) and fast solutions to otherwise intractable problems. Exact algorithms, on the contrary, ensure the achievement of optimal solutions but cannot deal with large and real instances of the problem. This is why a wide variety of heuristic strategies have been formulated, so as to obtain good quality solutions in an acceptable computational time. Some of the best-known level-oriented algorithms which are able to deal with guillotine cuttings are [17]: *First-Fit Decreasing Height*, *Next-Fit Decreasing Height*, *Best-Fit Decreasing Height*, and *SPLIT*. Many papers propose improvements to these level-based placement heuristics, or combine them with other approaches, such as genetic algorithms [1]. As a more general and sophisticated method, different types of hybrid algorithms and meta-heuristics have been considered [1, 9, 12, 18].

In spite of the great number of works dealing with this problem, only a few deal with real-world multi-objective constraints [13, 20]. These existing approaches for this multi-objective formulation of the 2DSPP are all based on the application of MOEAs. The cutting patterns provided by the approach presented in [4, 20] always allow for guillotinable cuts, while those achieved by [13] can only be non-guillotinable. Works dealing with guillotine cuttings [4, 20], which have been taken as a reference in our work, are based on a codification of solutions where the cutting layouts are directly represented through a post-fix notation. The approach proposed in [4] clearly improved the previous proposal presented in [20]. However, although the results obtained in [4] were competitive, not only compared to the other existing multi-objective approach [20], but also when compared to some single-objective algorithms [16], we realised that the search space defined by the chosen encoding scheme became intractable for large problem instances. For this reason, it is necessary to check other type of encoding schemes. In this sense, we have designed an intermediate alternative which combines the solution encoding with the usage of decoding heuristics. For this proposal we have considered the properties of the existing wide variety of heuristics for the 2DSPP and we have designed a hyperheuristic-based encoding scheme which may allow us to combine them in an successfully way. Such codification introduce coding of solutions, incorporating low-level placement heuristics to decode the solutions. In [5], we have solved the 2DSPP using this hyperheuristic-based encoding scheme. So, here, the study includes a new problem, 2DCSP, making a comparative and trying to obtain more global and general conclusions.

Focusing on the second problem - 2DCSP - a large number of exact algorithms [11, 21, 22] have been proposed to solve the single-objective formulation of it. There are also many heuristics to solve the problem. Most of these heuristics solve the non-guillotine version of the problem [2]. Only a few of them, deal with the guillotine case [17]. For this problem, we haven't found any related work dealing with multiple objectives, so we have taken as a reference the existing approaches for the multi-objective 2DSPP. Considering the similarities between both problems, we have also applied MOEAs and a hyperheuristic encoding scheme to solve the multi-objective 2DCSP. This way, we can check the efficiency of hyperheuristics when applied to problems with a wide solution space and a large number of specific-designed heuristics.

3. MULTI-OBJECTIVE OPTIMISATION

Multi-objective or multi-criteria optimisation problems (MOPs) [19] arise in most real-world disciplines. While in single-objective optimisation the optimal solution is usually clearly defined, this does not hold for MOPs. Instead of a single optimum, there is rather a set or front of alternative trade-offs, known as *Pareto-optimal* front, constituted by the non-dominated solutions y. These solutions are optimal in the sense that no other solutions in the search space are superior to them when all objectives are considered. The final aim when dealing with MOPs is to obtain a non-dominated solution set which, in the best case, will coincide with the Pareto-optimal front. From the resulting final solution set, a human decision maker will be able to select a suitable compromise solution.

For the guillotine problems here analysed, the maximisation of the total profit or the minimisation of total length, involves an optimal usage of the raw material, thus implying the generation of compact cutting patterns containing little internal trim loss. For real instances, filling all the internal gaps usually involves the location of a higher number of pieces. Usually a higher number of cuts is necessary to obtain a higher number of pieces. For this reason, we can state that both objectives have at least a certain degree of conflict, and so, we will obtain sets of non-dominated solutions from which final users will be able to choose.

Since exact approaches are practically unapproachable for most MOPs, a wide variety of (meta)-heuristic algorithms have been designed. Two common approaches for simplifying the MOPs solution are: convert the original problem into a single-objective one by combining or aggregating the multiple objectives into a single function; or, translate some of the objectives into constraints. Usually, a more appropriate approximation involves the application of techniques that can specifically deal with multiple objectives and MOPs intrinsic complexity (very large search spaces, uncertainty, noise, disjoint Pareto curves, etc.).

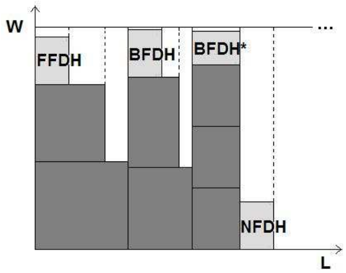

Figure 2: Heuristics

Evolutionary algorithms (EAs) have shown great promise for calculating solutions to large and difficult optimisation problems and have been successfully used across a wide variety of real-world applications. In fact, when applied to MOPs, EAs seem to perform better than other blind search strategies. Although this statement must be qualified with regard to the no free lunch theorems for optimisation, to date there are few, if any, alternatives to EA-based multi-objective optimisation. The use of EAs to solve problems of this special nature has been motivated mainly because they are able to capture multiple Pareto-optimal solutions in a single simulation run - which is possible thanks to their population-based feature - and to exploit similarities of solutions by recombination. EAs that are specifically designed to deal with multiple objective functions are known as MOEAs.When designing MOEAs two major problems must be addressed: how to accomplish fitness assignment and selection in order to guide the search towards the Pareto-optimal set, and how to maintain a diverse population in order to prevent premature convergence and achieve a well distributed trade-off front. Many alternatives have been proposed in an attempt to adhere to such design goals:VEGA, NPGA, NSGA [6], SPEA [26], IBEA [25], etc.

4. HYPERHEURISTIC-BASED ENCODING SCHEMES

A hyperheuristic is a heuristic search method that seeks to automate, often by the incorporation of machine learning techniques, the process of selecting, combining, generating or adapting several simpler heuristics (or components of such heuristics) to efficiently solve computational search problems. One of the motivations for studying hyperheuristics is to build systems which can handle classes of problems rather than solving just one problem.

As mentioned before, for both guillotine cutting problems, 2DCSP and 2DSPP, we can find a wide range of placement heuristics in the literature. Taking this into account and considering the properties of hyperheuristics, in this work, we propose a hyperheuristic encoding scheme which make the most of the existing heuristics, and at least takes into account the implicit features of the problem. We apply this encoding scheme to solve both problems, 2DCSP and 2DSPP. As hyperheuristics are a more general procedure for optimisation which deals with the process to choose/combine the right heuristics for solving the problem at hand, the used evolutionary operators are more general and could be used for other problems that are implemented using hyperheuristics.

4.1 Low-level heuristics

The first decision to implement a hyperheuristic-based approach lies on the definition of the set of placement heuristics to be combined. In the literature we can find a huge amount of heuristics for non-guillotine cutting problems, but the number of proposals for the guillotine case is not so extensive. From the existing heuristics we have selected four of the most-known. Originally, they all fix the orientation of the objects such that their width is not lower than their height. Then, objects are ordered from highest to lowest length. Once the order of the pieces to be packed is established, the heuristics must decide where to arrange a given object at a given moment, considering the open and still unfilled levels (see Figure 2). The selected heuristics are briefly described next:

- *Next Fit Decreasing Height* (NFDH) [3]: rectangles are packed left justified on a level until the next rectangle will not fit, in which case it is used to start a new level above the previous one, on which packing proceeds.

- *First Fit Decreasing Height* (FFDH) [3]: places each rectangle left justified on the first (lowest) level in which it will fit. If none of the current levels has room, then a new level is started.

- *Best Fit Decreasing Height* (BFDH) [16]: packs the next rectangle left justified on that level, among those that can accommodate it, for which the residual horizontal space is a minimum. If no level can accommodate it, a new one is created.

- *Best Fit Decreasing Height** (BFDH*) [1]: seeks to improve BFDH heuristic by allowing object rotations, so that when the algorithm searches to include the current object into a sub-area it tests both orientations. This heuristic is not applicable to the 2DCSP, which does not allow rotation of pieces.

4.2 Representations

For the 2DSPP an individual is represented by a sequence (PH, p, o, r), where PH is the identifier of the original low-level placement heuristic to be applied, p is the number of pieces that such a heuristic must arrange on the available material, o determines the order criterion and r is a rotation criterion. (see Figure 3). Chromosomes have a variable length j, which goes from $j = 1$ (there is one single sequence (PH, n, o, r) so that the same heuristic arranges all the available pieces) to $j = n$ (the n available objects are arranged on an independent way, i.e., $\forall i \in [1, n], p_i = 1$). Note that in a valid representation all the demanded objects must be arranged by the heuristics, i.e., $(\sum_{i=0}^{j} p_i) = n$. The basic order criteria used are: decreasing lengths, decreasing widths, decreasing areas and decreasing perimeters. The rotation criteria used are: width greater or equal than the lengths, lengths greater or equal than the widths, rotate no object and rotate all the objects. For the generation of the initial individuals, sequences (PH, p, o, r) are generated, until there are no more pending pieces to be arranged, where PH is randomly selected from the set of four used placement heuristics, p is randomly selected from the interval $[1, a_p]$ where a_p is the number of remaining or still available pieces, o is the order criterion randomly generated among the four available, and r is the rotation criterion randomly generated among the four available.

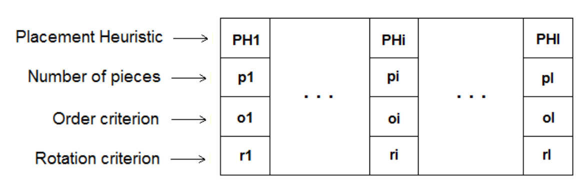

Figure 3: Representation based on hyperheuristics

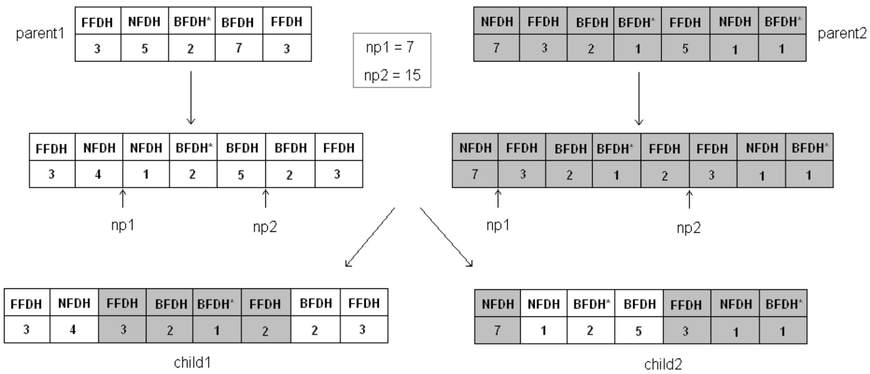

Figure 4: Two-point crossover

For the 2DCSP, we use a similar representation (PH, p, o), were PH, p and o have the same meaning as in the 2DSPP representation. In this problem the pieces cannot be rotated, so this criterion is not used and we cannot apply the BFDH* heuristic. For the generation of the initial individuals, sequences (PH, p, o) are generated, until there are no more pending pieces to be arranged. PH is randomly selected from the set of three used placement heuristics, p is randomly selected from the interval $[1, a_p]$ where a_p is the number of remaining or still available pieces, and o is the order criterion randomly generated among the four available.

4.3 Evaluation of Objectives

In these non-direct codifications, MOEAs evolve populations of individuals which represent hyperheuristics, so that, the chromosome must be interpreted to obtain the problem solution. The representation is analysed from left to right, applying for each sequence (PH, p, o, r) or (PH, p, o), the heuristic PH in order to locate the next p pieces. These pieces are arranged fulfilling with their corresponding terms of order and orientation. When pieces are located on the top of existing levels, verticals constructions are created. When a new level is opened at the right of the existing ones, a horizontal combination of the adjacent levels is generated. The decodification of the chromosome provides a problem solution represented by a post-fix notation of vertical and horizontal cuts. In case of 2DCSP, the decodification is done until no more pieces fit in the material. Then, such a solution can be evaluated in order to obtain the values of the two objectives: overall length and number of necessary cuts for 2DSPP, and total profit and number of necessary cuts for 2DCSP. For this purpose, the methods applied are based on the usage of stacks and the post-fix notation, which represents the chromosome [18]. For the evaluation of the second objective - the number of cuts - an iterative method is applied. The chromosome is traversed from left to right, interpreting every element and creating the indicated constructions, thus calculating the partial widths and lengths. At least one cut is necessary for each implied vertical or horizontal combination of pieces. If the combined rectangles do not match in length (for vertical builds) or in width (for horizontal builds), an extra cut is required for the construction. At the end of the process, the complete final pattern is obtained. In this case, the value of the first objective in 2DSPP - overall length - is immediately given by the length of the resulting final pattern. The value of the first objective in 2DCSP - total profit - is given by the sum of the profits of the pieces that can be located on the available surface.

4.4 Operators

Several crossover and mutation operators were designed and tested with these representations. Some of them are more general and others more specific to deal with certain constraints of the representation. After designing and testing these crossover operators, we detected that the one achieving better results is the two-point crossover which is based on the accumulated number of pieces for the points selection. This two-point crossover considers the number of accumulated pieces within the representation. So, two number of pieces np_1 and np_2 are randomly generated, such that $1 \leq np_1 < np_2 < n$. For each parent, it is necessary to find the chromosome positions where the number of arranged pieces sums np_1 and np_2. When the sum of the pieces do not coincide with the generated value np, the first sequence (PH_i, p_i, o_i) or (PH_i, p_i, o_i, r_i), satisfying $\sum_{x=0}^{i} p_x > np$ should be split into two different sequences (PH_j, p_j, o_j) and (PH_k, p_k, o_k) or (PH_j, p_j, o_j, r_j) and (PH_k, p_k, o_k, r_k), such that $PH_j = PH_k = PH_i$, $p_k = (\sum_{x=0}^{i} p_x) - np$, and $p_j = p_i - p_k$ (see Figure 4). Once the two cross points are determined inside both parents, the central pairs between them are exchanged among the two parents in order to generate the new off-springs.

From all the designed and tested mutation operators, the one showing a more promising behaviour is based on the application of three different type of movements inside the chromosome. Each of the following movements is applied upon the algorithm mutation probability p_m:

- *add*: randomly selects a sequence (PH_i, p_i, o_i) or (PH_i, p_i, o_i, r_i) inside the representation. Then, it generates a new sequence (PH, p, o) or (PH, p, o, r), where PH is a random selected heuristic, o is a random selected order criteria, r is a random selected rotation criterion, and p is a random number within the interval $[1, p_i]$. Then, p_i is updated with the value $p_i - p$. If after the update, $p_i \neq 0$, pairs in positions $i, \ldots, j$ are displaced one position to the right, increasing the total length of the individual ($j = j + 1$). Finally, the new sequence is introduced in position i.

- *remove*: randomly selects a sequence (PH_i, p_i, o_i) or (PH_i, p_i, o_i, r_i) inside the representation. If the selected sequence is the last one in the representation ($i = j$), then $p_{i-1} = p_{i-1} + p_i$. In other case, $p_{i+1} = p_{i+1} + p_i$ and the pairs in positions $i+1, \ldots, j$ are displaced one position to the left. In both cases, the length is updated ($j = j - 1$). This operation can be applied only if initially $j > 1$.

- *replace*: randomly selects a sequence (PH_i, p_i, o_i) or (PH_i, p_i, o_i, r_i) inside the representation. PH_i is randomly fixed to one of the defined low-level heuristics.

Table 1: Configuration for hyperheuristic encoding schemes

Problem	Algorithm	Crossover	Mutation	Population
2DSPP	NSGA-II	0.6	0.4	50
2DCSP	NSGA-II	0.7	0.3	50

Table 2: Single-objective low-level heuristics and multi-objective approach for 2DSPP

Solution	nice200_40		path200_40		nice500_5		path500_5	
Approach	length	cuts	length	cuts	length	cuts	length	cuts
NFDH	111	371	127	350	109	943	120	896
FFDH	107	374	114	364	106	946	113	904
BFDH	107	374	114	392	106	946	113	904
BFDH*	107	373	114	354	106	946	111	905
Hyp.	106.53	356.56	103.51	348.66	104.61	854.73	104.63	881.23
Encod. Scheme	114.33	336.30	115.81	318.90	109.07	816.55	135.11	830.96

Table 3: Single-objective low-level heuristics and multi-objective approach for 2DCSP

Solution	ATP33s		ATP37s		CL_07_100_08		Hchl5s	
Approach	profit	cuts	profit	cuts	profit	cuts	profit	cuts
NFDH	166780	15	271275	12	9506	2	27738	8
FFDH	166780	15	271275	12	9506	2	27738	8
BFDH	166780	15	271275	12	9506	2	27738	8
Hyp.	215027.70	30.50	357343.00	34.00	18964.00	25.06	38280.00	17.00
Encod. Scheme	60633.60	3.03	133168.80	4.00	9999.00	2.00	9936.00	3.00

5. EXPERIMENTAL EVALUATION

The experimental evaluation was performed on a dedicated Debian GNU/Linux cluster of 20 dual-core nodes. Each node consists of two Intel® Xeon 2.66 GHz and has 1GB RAM and a Gigabit Ethernet interconnection network. The framework and the approach for the problem were implemented in C++ and compiled with *gcc 4.1.3* and MPICH *1.2.7*. For the computational study, some test instances available in the literature [23] have been used for the 2DSPP, and others [7, 10] have been used for the 2DCSP. We use these benchmarks to compare our results with the existing results.

In order to avoid the implementation of the most widely used MOEAs, our approaches are based on METCO [15], a Parallel Plugin-Based Framework for Multi-Objective Optimisation. The framework provides implementations of MOEAs such as NSGA-II [6], SPEA2 [26], IBEA [25], etc. It allows the users to simply specify the details related to the problem (representation, evaluation of objectives, operators, ...) without having to worry about the internal details of the algorithm implementation. The framework also provides a simple, flexible, and efficient interface to setup and tune the parameters being used inside the algorithms. In order to perform more accurate parametrisation, the tool is able to run on parallel environments.

Inside METCO we have defined two individuals - one for the 2DSPP and other for the 2DCSP - using the hyperheuristic encoding scheme presented in section 4. For each problem, we have defined the corresponding representation, evaluation, generation, and operator methods involved. The approaches were evolved using three different MOEAs: NSGA-II, SPEA2, and an adaptive version of IBEA. Every pair algorithm-problem was tested using different sets of parameter configurations. In general, the hyperheuristic results were able to improve the heuristic ones, but, after performing an initial tuning, we have selected the pair algorithm-parameter better performing for each problem. The better performing algorithms and parameters, and so, used for the next experiments, are shown in Table 1. Note that mutation probabilities are relatively high because the mutation operator is not too aggressive. As in previous cutting related works [4, 20], NSGA-II clearly showed a better behaviour than the other tested algorithms. The results using them have been omitted due to lack of space.

The application of MOEAs that generate solutions to the 2DSPP and the 2DCSP according to two different optimisation criteria has a major advantage for potential customers: such approaches provide a set of solutions offering a range of trade-offs between the two objectives, from which clients can choose according to their needs, e.g. cost associated with the raw material or even times imposed for the production process. However, dealing with more than one optimisation objective does not necessarily imply a reduced solution quality at the expense of possibly optimising multiple objectives. On the contrary, for the problem 2DSPP studied, the direct encoding scheme approach has been checked and, by considering both, the number of cuts and the length, the approach derived solutions with wastage levels similar to most previous approximations which just seek to optimise the overall length [4]. For larger problems, where a high amount of pieces must be arranged, the search space of solutions is too large, and so, the single-objective approaches which seek to optimise the overall length obtain better results. For this reason, we have designed an encoding scheme which can deal with a more reduced search space for the considered cutting problems, so that solutions with quality comparable to those achieved by the single-objective approaches can be obtained.

Following such a goal, we have designed a hyperheuristic-based encoding scheme which combines different existing low-level heuristics in order to improve the solutions given when the heuristic methods are individually applied. Table 2 and Table 3 show the values of the two optimisation objectives when the low-level heuristics are individually applied and when the proposed multi-objective ap-

Table 4: Optimal and Approximated Solutions

Instance	Single-objective Solution		Hyp. Approach	
	profit	cuts	profit	cuts
ATP33s	236611	34	215027.70	30.50
			60633.60	3.03
ATP37s	387276	39	357343.00	34.00
			133168.80	4.00
CL_07_50_09	22088	14	13129.00	12.00
			9506.00	2.00
CL_07_100_08	22443	30	18964.00	25.06
			9999.00	2.00
Hchl2	9954	21	8992.00	18.00
			1800.00	3.00
Hchl5s	45410	31	38280.00	17.00
			9936.00	3.00

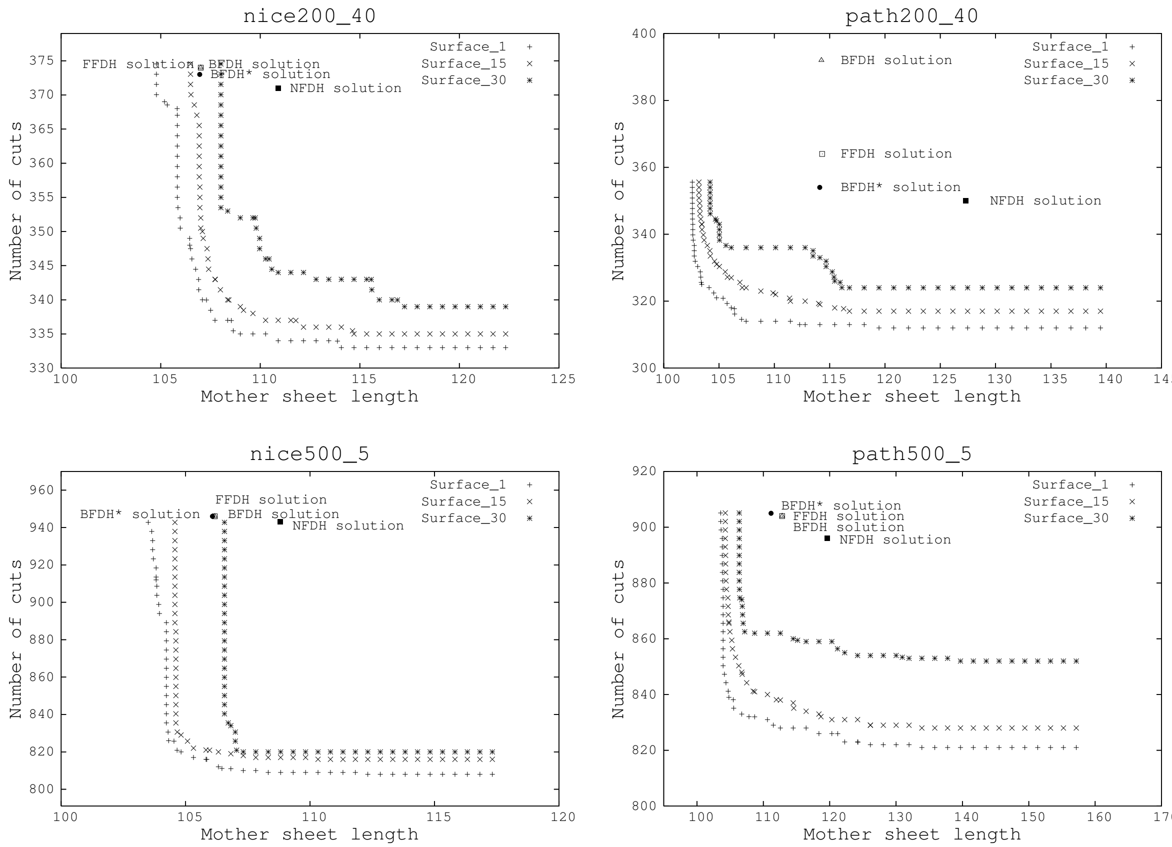

Figure 5: Attainment surfaces for 2DSPP hyperheuristic encoding scheme

proach is applied for both, 2DSPP and 2DCSP, respectively. Tables show results for four different problem instances. Both problems have been tested with many more instances, but for reasons of space we only shows a set of them, from the smallest to the most complex ones. Note that in the case of the 2DCSP the BFDH* heuristic is not applicable. For the hyperheuristic approaches, thirty executions are performed, using the algorithm and configuration parameters given in Table 1. The stop criterion is fixed to 30 minutes for the 2DSPP and to 10 minutes for the 2DCSP. For each repetition, two solution points are selected: the lower length or higher profit one - depending on the problem - and the lower number of cuts one. The average values are shown for the lowest-length or highest-profit solution (first row of the encoding schemes) and the lowest-cuts solution (second row).

Analysing Table 2, we can see that the hyperheuristic approach is able to improve - for both objectives - the values obtained by the heuristics, i.e, by the application of hyperheuristics principles and a multi-objective approach, we have improved the solutions obtained by tailor-made algorithms for the single-objective 2DSPP. On the other hand, we can analyse Table 3 to study the results for the 2DCSP. As we can notice, for each test problem instance the three single-objective heuristics provide the same solution, and results given by the hyperheuristic approach improve these values, i.e., combining the heuristics we have obtained better results.

Unlike in case of the 2DSPP, for the 2DCSP the single-objective optimal solution is known for the selected instances. In Table 4 the optimal single-objective solutions of the 2DCSP instances are compare to the solutions given by the multi-objective approach. As in the previous tables, here two different solutions are shown for each pair instance-approach: the average best solution when considering the profit objective, and the average best solution when attending to the number of cuts objective. Looking at the results we realise that the hyperheuristic approach is not able to reach the single-objective optimal profit values, although it achieves solution with profit values rather close to the optimal profit, but using lower number of cuts. It is important to note that the applied hyperheuristic is based on single-objective heuristics which don't provide high quality results for this problem because originally, they have been created to deal with level-oriented cutting problems.

Until the moment, we have just analysed the solutions obtaining the extreme values for each of the objectives (length/profit and cuts). Now, it is necessary to compare the complete set of solutions obtained by the encoding schemes. We would like to clearly identify the search space area being explored. Directly plotting Pareto fronts could be rather messy since we are dealing with the results of thirty executions, so as an alternative we have used attainment surfaces [8]. An attainment surface is the family of tightest goals that has been attained by the approximation set defining it. Using attainment surfaces, it is much easier to identify "gaps" in the distribution of points. If we want to display the outcome of multiple runs of one or more optimisers instead of plotting one front for each of the executions, we can make use of the summary attainment surfaces [14]. Note that if we have performed n different runs, the summary attainment surface s weakly dominates summary attainment surfaces $s+1, s+2, \dots n$. Summary attainment surface plots are easier to interpret than plots of many result surfaces since summary surfaces never cross each other.

Figure 5 shows, for four different instances, the summary at-

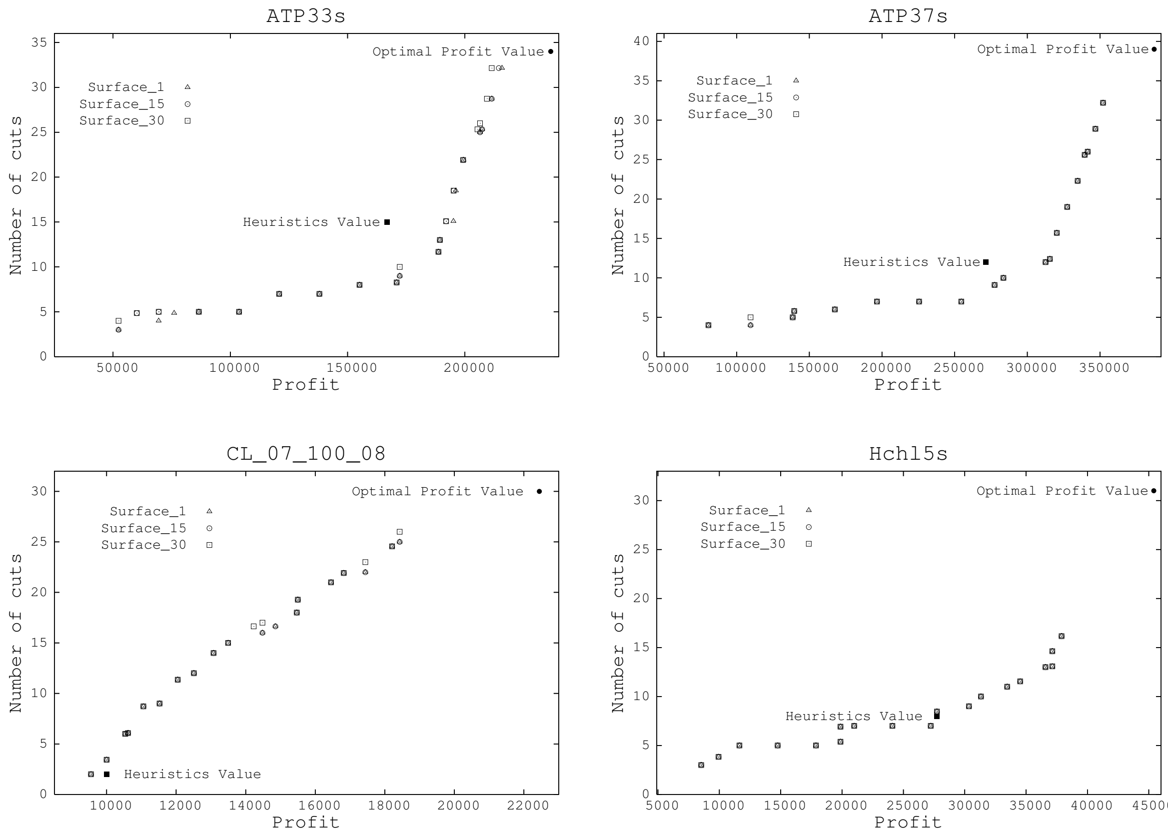

Figure 6: Attainment surfaces for 2DCSP hyperheuristic encoding scheme

tainment surfaces 1, 15, and 30, obtained by the hyperheuristic approach when applied to the 2DSPP. The approximate solutions given by each individual heuristic are also attached. The hyperheuristic encoding scheme shows a good compromise between the objectives, and the attainment surfaces have many points which dominate the approximate solution given by the individual heuristics. Similar results are shown in Figure 6 for the 2DCSP. In this case, the solutions optimising the profit objective are also included in the figure. We can note that the hyperheuristic approach is not able to reach the single-objective optimal profit values. There are no points which dominates the exact values, but the hyperheuristic approach provides profit values rather close to the optimal profit, using a number of cuts much less than the number of cuts corresponding to the optimal profit. Moreover, it is important to note that the implementation of a tailor-made exact algorithm involves a certain difficulty and a huge computational cost, unlike the hyperheuristic, which is a fairly general implementation for guillotine cutting problems. Anyway, the hyperheuristic is able to provide competitive results when compare to the individual heuristics, as happened with the 2DSPP. As an additional advantages of the multi-objective approaches, we provide a set of solutions from which a human decision maker is able to select the suitable compromise final solution.

6. CONCLUSIONS

Real-world multi-objective formulations of the 2DSPP and 2DCSP have been presented. The two objectives considered for the 2DSPP were minimise the overall length of the raw material and the total number of cuts needed to obtain the complete set of demanded pieces. The two objectives considered for the 2DCSP were maximise the total profit and minimise the total number of cuts to obtain the total demanded pieces. The advantage of having a multi-objective approach is to capture multiple Pareto-optimal solutions in a single simulation run, so that, then, a human decision maker will be able to select a suitable compromise solution. Many approximations to the single-objective formulation of the problems appear in the literature, but the number of multiple-objective approaches for the guillotine 2DSPP is very reduced, or even unknown for the guillotine 2DCSP. We have chosen the NSGA-II algorithm and we have proposed a new type of codification based on the combination of different low-level heuristics. This way, we can reduce the search space with regard to the search space using a post-fix notation, which is specially necessary to solve the larger test problem instances of the multi-objective 2DSPP. Moreover, we obtain a quite general implementation for guillotine cutting problems. Results demonstrate that the 2DSPP and 2DCSP hyperheuristic encoding schemes are able to obtain competitive solutions when compared to single heuristics. Even more, it achieves solutions which completely dominate the ones obtained by the single heuristics. However, when we use hyperheuristic encoding scheme for the 2DCSP, the approach is not able to reach the optimal profit values. Nevertheless, it provides profit values rather close to the optimal profit, using a number of cuts much less than the number of cuts corresponding to the optimal profit.

In 2DCSP case we have the exact values for the test problem instances and for the 2DSPP we only have the approximate values given by the individual heuristics. So, as future work, it would be interesting to implement an exact algorithm for the 2DSPP to obtain the exact solutions of the here used test problem instances, and to check the 'real' competitiveness of the 2DSPP hyperheuristic approach. Moreover, we think that an important line of future research lies on the design of other heuristics to add them to the hyperheuristic encoding scheme for the 2DCSP. Finally, we could compare results obtained using these hyperheuristics with other encoding scheme results, for example, using a direct encoding scheme for both problems.

7. ACKNOWLEDGMENTS

This work was funded by the EC (FEDER) and the Spanish Ministry of Science and Technology as part of the 'Plan Nacional de I+D+i' (TIN2008-06491-C04-02). The Canary Government has also funded this work through the PI2007/015 research project. The work of Jesica de Armas was funded by grant FPU-AP2007-02414.

8. REFERENCES

[1] A. Bortfeldt. A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. *European Journal of Operational Research*, 127(3):814–837, 2006.

[2] E. K. Burke, G. Kendall, and G. Whitwell. A New Placement Heuristic for the Orthogonal Stock-Cutting Problem. *Operations Research*, 52(4):655–671, 2004.

[3] E. Coffman, M. Garey, D. Johnson, and R. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9:808–826, 1980.

[4] J. de Armas, C. León, G. Miranda, and C. Segura. Optimisation of a multi-objective two-dimensional strip packing problem based on evolutionary algorithms. *International Journal of Production Research*, 48(7):2011–2028, 2009.

[5] J. de Armas, G. Miranda, and C. Leon. Hyperheuristic Codification for the Multi-Objective 2D Guillotine Strip Packing Problem. In *IEEE Congress on Evolutionary Computation*, pages 2883–2890. IEEE Computer Society, July 2010.

[6] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *VI Conference on Parallel Problem Solving from Nature*, volume 1917 of *LNCS*, pages 849–858. Springer, 2000.

[7] DEIS - Operations Research Group. Library of Instances: Bin Packing Problem. http://www.or.deis.unibo.it/research_pages/ORinstances/2CBP.html.

[8] C. Fonseca and P. J. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In *Parallel Problem Solving from Nature*, LNCS, pages 584–593. Springer-Verlag, 1996.

[9] P. Garrido and M. C. Riff. Collaboration Between Hyperheuristics to Solve Strip-Packing Problems. In *Foundations of Fuzzy Logic and Soft Computing*, volume 4529 of *LNCS*, pages 698–707. Springer, 2007.

[10] M. Hifi. 2D Cutting Stock Problem Instances. ftp://cermsem.univ-paris1.fr/pub/CERMSEM/hifi/2Dcutting/.

[11] M. Hifi. An Improvement of Viswanathan and Bagchi's Exact Algorithm for Constrained Two-Dimensional Cutting Stock. *Computer Operations Research*, 24(8):727–736, 1997.

[12] E. Hopper and B. C. H. Turton. A Review of the Application of Meta-Heuristic Algorithms to 2D Strip Packing Problems. *Artificial Intelligence Review*, 16(4):257–300, 2001.

[13] S. Illich, L. While, and L. Barone. Multi-objective strip packing using an evolutionary algorithm. In *IEEE Congress on Evolutionary Computation*, pages 4207–4214, September 2007.

[14] J. Knowles. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, pages 552–557. IEEE Computer Society, 2005.

[15] C. León, G. Miranda, and C. Segura. METCO: A Parallel Plugin-Based Framework for Multi-Objective Optimization. *International Journal on Artificial Intelligence Tools*, 18(4):569–588, 2009.

[16] C. L. Mumford-Valenzuela, J. Vick, and P. Y. Wang. *Metaheuristics: computer decision-making*, chapter Heuristics for large strip packing problems with guillotine patterns: an empirical study, pages 501–522. Kluwer Academic Publishers, 2004.

[17] N. Ntene and J. Van Vuuren. A survey and comparison of guillotine heuristics for the 2D oriented offline strip packing problem. *Discrete Optimization*, 6(2):174–188, January 2009.

[18] T. Ono and T. Ikeda. Optimization of two-dimensional guillotine cutting by genetic algorithms. In H. J. Zimmermann, editor, *European Congress on Intelligent Techniques and Soft Computing*, volume 1, pages 7–10, 1998.

[19] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, New York, 1986.

[20] S. Tiwari and N. Chakraborti. Multi-objective optimization of a two-dimensional cutting problem using genetic algorithms. *Journal of Materials Processing Technology*, 173:384–393, 2006.

[21] K. V. Viswanathan and A. Bagchi. Best-First Search Methods for Constrained Two-Dimensional Cutting Stock Problems. *Operations Research*, 41(4):768–776, 1993.

[22] P. Y. Wang. Two Algorithms for Constrained Two-Dimensional Cutting Stock Problems. *Operations Research*, 31(3):573–586, May-June 1983.

[23] P. Y. Wang and C. L. Valenzuela. Data set generation for rectangular placement problems. *European Journal of Operational Research*, 134(2):378–391, October 2001.

[24] G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109–1130, December 2007.

[25] E. Zitzler and S. Künzli. Indicator-Based Selection in Multiobjective Search. In *VIII Conference on Parallel Problem Solving from Nature*, volume 3242 of *LNCS*, pages 832–842. Springer, 2004.

[26] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Evolutionary Methods for Design, Optimization and Control*, pages 19–26, Barcelona, Spain, 2002. CIMNE.