

Data mining based framework to assess solution quality for the rectangular 2D strip-packing problem

Alvaro Neuenfeldt Júnior*, Elsa Silva, A. Miguel Gomes, Carlos Soares, José Fernando Oliveira

INESC TEC, Faculty of Engineering, University of Porto, Portugal



ARTICLE INFO

Article history:

Received 7 November 2017

Revised 14 September 2018

Accepted 3 October 2018

Available online 4 October 2018

Keywords:

Strip-packing problem

Cutting and packing problem

Knowledge discovery

Data mining

Heuristics

Regression analysis

ABSTRACT

In this paper, we explore the use of reference values (predictors) for the optimal objective function value of hard combinatorial optimization problems, instead of bounds, obtained by data mining techniques, and that may be used to assess the quality of heuristic solutions for the problem. With this purpose, we resort to the rectangular two-dimensional strip-packing problem (2D-SPP), which can be found in many industrial contexts. Mostly this problem is solved by heuristic methods, which provide good solutions. However, heuristic approaches do not guarantee optimality, and lower bounds are generally used to give information on the solution quality, in particular, the area lower bound. But this bound has a severe accuracy problem. Therefore, we propose a data mining-based framework capable of assessing the quality of heuristic solutions for the 2D-SPP. A regression model was fitted by comparing the strip height solutions obtained with the bottom-left-fill heuristic and 19 predictors provided by problem characteristics. Random forest was selected as the data mining technique with the best level of generalisation for the problem, and 30,000 problem instances were generated to represent different 2D-SPP variations found in real-world applications. Height predictions for new problem instances can be found in the regression model fitted. In the computational experimentation, we demonstrate that the data mining-based framework proposed is consistent, opening the doors for its application to finding predictions for other combinatorial optimisation problems, in particular, other cutting and packing problems. However, how to use a reference value instead of a bound, has still a large room for discussion and innovative ideas. Some directions for the use of reference values as a stopping criterion in search algorithms are also provided.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

The two-dimensional strip-packing problem (2D-SPP) consists of packing a set of small rectangular rectangles into the strip of fixed width and virtually infinite height, minimising the required height. The small rectangles must be positioned without any overlap between them and completely inside the strip. According to the typology of Wäscher, Haußner, and Schumann (2007), this description fits in the definition of cutting and packing problems as a 2D rectangular Open Dimension Problem (ODP). In the 2D-SPP tackled in this work, the rectangles should be orthogonally packed inside the strip and are allowed to rotate 90 degrees. Typical 2D-SPP applications are found in manufacturing, including the cutting of metals, textiles or paper rolls.

The 2D-SPP is a NP-hard problem and can be solved either by exact methods or heuristic approaches (Alvarez-Valdés, Parreño, & Tamarit, 2008; Martello, Monaci, & Vigo, 2003). Exact methods are based on mathematical programming models and are able to obtain optimal solutions or, at least, determine the solution quality. However, these methods are not able to tackle the complexity and size of real-world applications. Instead, heuristic approaches have been used in these situations, given the low computational times required when compared to exact methods. However, heuristic approaches do not guarantee optimality and do not provide any information about solution quality (Hopper & Turton, 2001b; Ntene & van Vuuren, 2009).

Lower bounds have been used to overcome this limitation in various cutting and packing problems (Fekete & Schepers, 2001; Lodi, Martello, & Monaci, 2002; Martello et al., 2003). However, almost none of the lower bounds available in the 2D-SPP literature allows rectangles to rotate 90 degrees, meaning that they cannot be used for solving the problem tackled in this work, as the lower bound can be greater than the optimal solution. The only exception

* Corresponding author.

E-mail addresses: alvaroj.eng@gmail.com (A. Neuenfeldt Júnior), emasilva@inesctec.pt (E. Silva), agomes@fe.up.pt (A.M. Gomes), csouares@fe.up.pt (C. Soares), jfo@fe.up.pt (J.F. Oliveira).

is the area lower bound proposed by Martello et al. (2003), which is based only on the area of all rectangles and the strip width. The main problem is the lack of accuracy of the area lower bound, mainly in problem instances where the optimal solution has waste space between the rectangles.

In this paper, we will tackle this decision-making problem from an expert system perspective. Although there is no definite definition for expert systems, and the meaning has even evolved, from the very beginning that expert systems are associated to the capacity of reasoning based on a body of knowledge, in opposition to the procedural framework. With the more recent availability of enormous amounts of data and the existence of a computational power to treat it, a new family of expert systems that replace the traditional “if-then rules” to represent knowledge but the knowledge extraction from the data itself, has arisen, grown and established itself. It is in this stream of expert systems that our work is inserted.

The main objective of this research is then to propose a data mining-based framework capable of assessing the quality of heuristic solutions for the 2D-SPP with 90° rotations, comparing the strip height of a given solution with a prediction of the height required to pack all rectangles into the strip. The prediction is obtained by fitting a regression model with data mining techniques. Besides providing a quality measure to heuristic solutions, the framework can also be used to develop more precise stopping criteria in local search algorithms with the goal of avoiding long computational times. Traditional stopping criteria rely on a high number of iterations to ensure solution quality. The proposed data mining-based framework can be extended to other cutting and packing problems, such as the bin packing problem (minimise number of used bins), the knapsack problem (maximise the total value in the knapsack), and the cutting stock problem (minimise the amount of scrap).

The framework was developed around a data mining approach, in which the predictors (explanatory variables) were developed according to relevant characteristics of the problem in order to provide an adequate way of measuring a known response variable for predefined observations (or problem instances). In the 2D-SPP, this known response variable is related to the strip height, which is calculated using constructive heuristics and local search algorithms. The data mining approach uses the predictors and the known response variable to fit a regression model with the goal of predicting the response variable for new problem instances. Different data mining techniques were tested to fit the regression model and to ensure an adequate generalisation level and the predictions' accuracy for new problem instances.

To illustrate the problem and the potential use of the predictions obtained by regression models fitted with the data mining-based framework proposed in this research, three examples comparing the predictions with optimal solutions and area lower bound values are presented.

Firstly, the problem instance *gcut1* proposed by Beasley (1985b) is presented (Fig. 1a). The wasted space found in the strip for the optimal solution (6%) is one of the largest for benchmark problem instances. The distance from the prediction ($\hat{H}^{ref} = 710$) to the optimal solution ($OS = 696$) is lower when compared with the distance between the area lower bound ($L_0 = 655$) and the optimal solution. Although \hat{H}^{ref} is higher than OS , \hat{H}^{ref} is more realistic and less optimistic when compared with the L_0 , which cannot feasibly be reached by any local search.

The second example is the problem instance *pt10_23_25* (Fig. 1b), generated using the 2DCPackGen (Silva, Oliveira, & Wäscher, 2014). The prediction ($\hat{H}^{ref} = 89$) is near the optimal solution ($OS = 88$). The area lower bound ($L_0 = 74$) is too optimistic, 16% below the optimal solution. The total wasted space available in the optimal solution is 16%, which is higher than the optimal waste

of problem instance *gcut1*. Finally, Fig. 1c shows an extreme toy example for a narrow strip where the total wasted space available in the optimal solution is 33% of the total area. The area lower bound ($L_0 = 530$) is very different when compared with the optimal solution ($OS = 700$). When compared with the area lower bound, the use of the prediction ($\hat{H}^{ref} = 620$) as stopping criterion is recommended.

The paper is organised as follows. Section 2 presents the literature review about the 2D-SPP and similar data mining approaches for cutting and packing problems. Section 3 presents the data mining-based framework. Section 4 focuses on the problem knowledge. Section 5 presents the data mining approach. Section 6 presents an assessment of the framework performance. Finally, Section 7 presents some final remarks and future research ideas. Additionally, in Appendix A we introduce basic notation used in this research for quick reference.

2. Literature review

The 2D-SPP is one of the most explored cutting and packing problems in real-world contexts.

In Oliveira, Neuenfeldt Júnior, Silva, and Carravilla (2016), an overview on heuristics for the 2D-SPP was extensively explored. The relevant literature was reviewed and links between the most frequently used heuristics and the 2D-SPP's characteristics were investigated. Most of the works published in recent years present successful heuristic approaches to the 2D-SPP with 90° rotations, while some also present heuristic approaches developed to tackle specific real-world problems. Validation of these approaches is usually done by comparison with several “standard” benchmark problem instances. This means that there are no types of absolute measure regarding the quality of solutions achieved by the proposed heuristic approaches.

A typical measure to assess solution quality in NP-hard combinatorial optimisation problems is the usage of lower bounds as a reference value. However, that it is not a viable option for the 2D-SPP with 90° rotations due to the lack of strong lower bounds. The only viable option is the area lower bound from Martello et al. (2003).

To access the quality of the solutions, two heuristics (Neveu, Trombettoni, Araya, & Riff, 2008; Wei, Zhang, & Chen, 2009) adopted non-zero-waste benchmark problem instances from datasets *gcut* (Beasley, 1985b), *cgcut* (Christofides & Whitlock, 1977), *ngcut* (Beasley, 1985a) and *bwmv* (Berkey & Wang, 1987). An interesting fact observed in both studies is the peculiar usage of lower bounds. Wei et al. (2009) presents a least wasted first heuristic, which uses the lower bounds proposed by Martello et al. (2003) in the non-rotated 2D-SPP as a reference value to measure the quality of the solutions. Neveu et al. (2008) proposed a method that adjusts the rectangles' positioning in the strip based on the location of the maximum holes. As in Wei et al. (2009), the solutions are evaluated using the lower bounds in the non-rotated 2D-SPP of Martello et al. (2003). Lower bounds that do not allow rectangles to rotate should not be used as reference values in the rotated case, since the optimal solution could be better. As examples of heuristics developed for solve the 2D-SPP with 90 degrees rotation using the area lower bound, He, Wu, and De Souza (2012) proposed an efficient deterministic heuristic algorithm that contains a rapid constructive phase and a partial tree search phase, working well mainly for zero-waste problem instances. In Özcan, Kai, and Drake (2013) a evolution of the best-fit position-based constructive heuristic is presented, improving the solutions for most of the benchmark problem instances.

Data mining approaches in cutting and packing problems are related with the process of converting problem information into

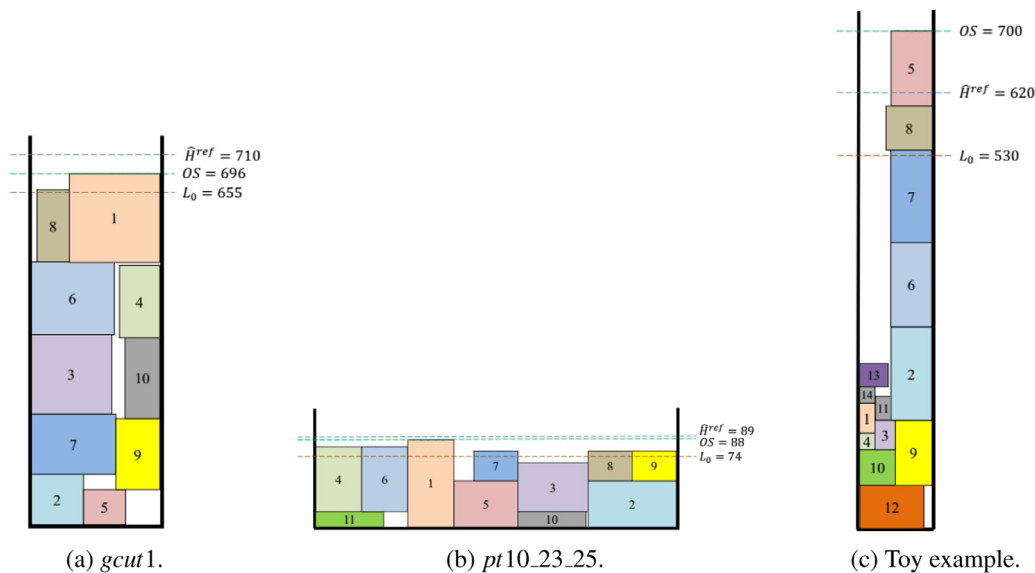


Fig. 1. Examples of height prediction (\hat{H}^{ref}), area lower bound (L_0), and optimal solution (OS).

measurable factors in order to reflect the main problem characteristics and compare the algorithms performance with different types of problem instances. However, the literature combining knowledge discovery with data mining techniques within the context of cutting and packing problems is not extensive. A literature review about the main contributions in this research field is presented in the next paragraphs. These contributions were fundamental to develop the data mining framework proposed in this research to predict the strip height.

The work by Smith-Miles and Lopes (2012) was one of the first to provide a comprehensive understanding about the methodologies used to measure the difficulty of problem instances. An extensive study on the most relevant characteristics to measure algorithm performance was conducted for six different combinatorial optimisation problems (assignment, travelling salesman, knapsack, bin-packing, graph colouring, and timetabling).

A framework to compare the strengths and weaknesses of optimisation algorithms was described by Smith-Miles, Baatar, Wrexford, and Lewis (2014), starting with the problem instance selection (or generation), feature selection, dimension reduction using principal component analysis and, finally, the development of algorithm performance metrics. A case study involving the graph colouring problem was presented. This framework is an improvement of the methodology proposed by Rice (1976) for the algorithm selection problem in order to predict the performance level based on characteristics found in the problem instances.

To replace an intuitive process in order to select the most important features of the irregular two-dimensional bin packing problem, López-Camacho, Terashima-Marín, and Ross (2010) defined a problem-state representation to improve the performance of a hyper-heuristic algorithm. In López-Camacho, Terashima-Marín, Ochoa, and Conant-Pablos (2013), a principal component analysis was used as a knowledge discovery method to understand the 1D and the 2D irregular bin packing problems' structure and its relation with the heuristics' performance. All data obtained was used as input to develop a unified hyper-heuristic (López-Camacho, Terashima-Marín, Ross, & Ochoa, 2014) capable of adapting its behaviour to each problem instance.

Simplifying the problem structure using linear correlation analysis was proposed in Santoyo, Ortega, Vargas, and Reyes (2015) in order to characterise the difficulty of the problem instances for the bin packing problem. A linear correlation analysis was conducted

in order to reduce a total of 27 features selected from the literature to only five metrics and used to compare the quality of six algorithm solutions. Perez, Frausto, Cruz, Fraire, and Santiago (2004) incorporated machine learning techniques in order to predict the performance of seven heuristic algorithms for new problem instances in the bin packing problem. A set of critical characteristics from solved problem instances was used as input data. A comparison between the augmented neural network and minimum bin slack heuristic for the one-dimensional bin packing problem was proposed in Almeida and Steiner (2016), incorporating characteristics from problem instances found in the literature.

For the 0–1 knapsack problem, Hall and Posner (2007) used a set of computed characteristics in order to develop a methodology for predicting the best procedure (branch-and-search or dynamic programming algorithm) that should be used for each particular problem instance. A regression model was fitted to predict the relative performance, considering six typical attributes found in the knapsack problem (problem size, the characteristics of rectangle value and size, the relationship between rectangle value and rectangle size, knapsack capacity and characteristics of the linear relaxation solution). These attributes were extracted to ten measures, incorporated into the regression model.

A small summary of the strengths and weaknesses of each data mining approach in cutting and packing problems is presented in the Table 1.

3. Data mining-based framework

This section focuses on introducing and describing the proposed framework (Fig. 2). The remainder of this section presents the concepts on each framework step.

The data mining-based framework was developed around the processing of two datasets used to fit and validate the regression model, which can be applied to predict the strip height in new problem instances. The Dataset (in blue) was used to develop the regression analysis and to fit the regression model, which is capable of predicting the strip height. The Validation dataset (in orange) was generated to perform adjustment evaluation tests on the regression model fitted with the Dataset, verifying the level of generalisation and predictions' accuracy in different problem instances. Finally, new problem instances (in green) can be solved by different local search algorithms, using as stopping criterion the

Table 1

Summary of the strengths and weaknesses of different data mining approaches.

Reference	Strength	Weakness
Smith-Miles and Lopes (2012)	For different cutting and packing problems, characteristics in the problem instances were found to measure the performance of different optimization algorithms.	The quantity and quality of the meta-data available must be sufficient to characterize a problem, which demands the generation of a relatively high diversity of problem instances.
Smith-Miles et al. (2014)	Provide a complex visual tool to verify the instance diversity and observe the strengths and weaknesses of different algorithms' performance predictions and the relative algorithm power across the instances space.	The instances space were performed in a two-dimensional plane, which may compromise the algorithm's ability to recognize some characteristics of specific problem instances.
López-Camacho et al. (2010)	A non intuitive scheme to define a numerical vector representation for an instance of a given problem is presented, capable to be used to develop hyper-heuristics for combinatorial optimization problems.	The proposed scheme does not consider the processing time consumed to verify the solutions' quality obtained for a single-heuristic or a hyper-heuristic and use only one improvement heuristic to find better solutions.
López-Camacho et al. (2013)	A structured method using principal component analysis to reduce the problem complexity, to gain a deeper understanding of the bin packing problems and the performance of heuristic approaches.	The main component analysis was applied in two steps, obtaining only two components to characterize the problem, where the variance explained combining the variance of these two components is 65%. It is necessary to consider feature combinations and their interactions to have a clearer insight of the prediction performance.
(López-Camacho et al., 2014)	A deterministic evolutionary hyper-heuristic methodology capable of producing quick good-quality solutions for 1D/2D bin packing problems was developed, not using any additional parameter for tuning and without the cost of trying all heuristics' portfolio.	As mentioned in the paper, a topic to be explored would be to develop a machine learning algorithm to select the most appropriate heuristic, rather than choosing the closest point to a particular problem state and then applying labeled heuristics.
Santoyo et al. (2015)	Common features in complex bin packing problem instances that can be incorporated to develop more effective and efficient algorithms.	The representation of the problem instances based on the characteristics referred as the most relevant to the problem is not clearly identified.
Perez et al. (2004)	The problem instances were clustered using k-means method, to create similar sets of instances groups to predict the algorithms' performance.	Only C4.5 method was used as machine learning technique to find the relation between problem characteristics and algorithms' performance. Lack of cross-validation steps to fit more accurate classification models.
Almeida and Steiner (2016)	Indication that the performance of the minimum bin slack heuristic may be greater than the augmented artificial neural networks in both solution quality and computational time for the bin packing problem.	The comparative considered the use of only two techniques, and some seminal characteristics of the problem were not described, as the dimensionality, items and object shape, and if 90° rotation was or not allowed. Small variability in the problem instances' characteristics.
Hall and Posner (2007)	Faster and cheaper identification of effective solution procedures, as well as an improved understanding of the relationship between problem characteristics and the performance of various procedures, with a good algorithm selection accuracy value (74%).	The proposed methodology needs to be tested in situations where a larger set of optimization algorithms are being selected.

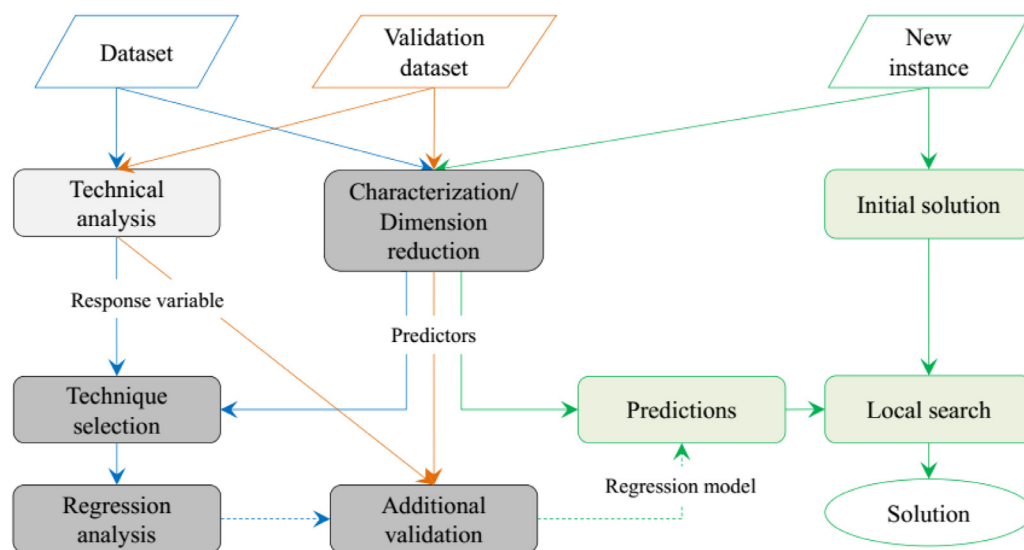


Fig. 2. Data mining-based methodological framework. Each step is presented in the following sections: Dataset (Section 4.1), Technical analysis (Section 4.2), Characterization and dimension reduction (Section 5.1), Technique selection (Section 5.2), Regression analysis (Section 5.3), and Validation dataset and additional validation (Section 5.4). (For interpretation of the references to colour in the text, the reader is referred to the web version of this article.)

strip height predictions provided by the regression model deployed after the additional validation step. Dashed arrows represent the transfer of a fitted regression model from one stage of the framework to another, and solid arrows are related with the transfer of input data (variables and datasets) to fit into or to use in the regression model. The knowledge discovery step is identified in light

grey, and the data mining approach steps are identified in dark grey.

A framework's cornerstone is to have a deep knowledge about the problem. This can be achieved through a 2D-SPP literature review (Neuenfeldt Júnior, Silva, Miguel Gomes, & Oliveira, 2017; Oliveira et al., 2016). The Problem knowledge (Section 4) was di-

vided into two steps: the generation of problem instances to be used as Dataset, and the technical analysis to describe the heuristics applied in order to obtain solutions for each problem instance. These solutions and the area lower bound were used as reference to define the known response variable for the regression analysis.

The problem instances of Dataset (Section 4.1) must be able to describe the wide quantity of characteristics and attributes of the strip-packing space, exploring the problem in different perspectives (Smith-Miles et al., 2014). The key is to avoid the problem representation with imprecise or biased information, and at the same time to include different characteristics and perspectives of the problem.

The Technical analysis (Section 4.2) requires the calculation of feasible solutions and a lower bound in each problem instance. As mentioned before, the heuristics have been used in real size problems, and are generally divided into constructive heuristics and improvement procedures (Oliveira et al., 2016). In constructive heuristics, the rectangles are successively positioned inside the strip until the last rectangle, and a complete solution is generated. In the improvement procedure (specifically for the local search), a solution, which can be obtained using some constructive heuristic, is improved by applying consecutive changes to the initial input sequence or to the arrangement of the rectangles in the strip. The local search runs until a stopping criterion is reached (Hopper & Turton, 2001a).

In the best-case scenario, the reference value should be the value of the optimal solution. However, with the exception of zero-waste problem instances, the value of the optimal solution is not known. Lower bounds have been used to bridge this gap.

Generally, the lower bounds are developed around mathematical integer formulations or problem relaxations. In the 2D-SPP, the most common relaxation is the one-contiguous bin packing problem (Alvarez-Valdes, Parreno, & Tamarit, 2009). Another possibility is to consider the relationship between the geometrical characteristics of the rectangles and the strip, transforming the rectangle dimensions and the strip's fixed dimension using the dual feasible function or calculating the total area that the rectangles can occupy inside the strip (Boschetti & Montaletti, 2010; Martello et al., 2003).

After the problem knowledge, in the Data mining approach (Section 5), the development of predictors that fully characterise the problem can be achieved by using concepts found in different cutting and packing problems, not only in the 2D-SPP. In addition, the study of how the problem instances were created and of its generation parameters is an important source of information. In general, the characterization considered in the development process is related to an exploratory context, considered in Smith-Miles and Lopes (2012) and Smith-Miles et al. (2014) as something similar to an art, combining multidisciplinary knowledge about the problem. Each selected feature must be relevant to the problem, avoiding spurious situations or measures that require long or non-feasible processing times to extract information from the problem instances (Hall & Posner, 2007).

The set of predictors selected is incorporated into a high-dimensional space, which can be reduced in order to facilitate the characterisation of the problem. Combinatorial optimisation approaches are useful in reducing the problem dimensions until a feasible and desirable subset of important predictors is reached, from a set of original descriptive variables in the problem dimensions, given different approximation criterion (Cadima, Cerdeira, & Minhoto, 2004). Another possibility is the use of mathematical dimension reduction techniques as principal component analysis or factorial analysis to produce uncorrelated components, retaining most of the information available in the predictors (López-Camacho et al., 2013). For example, the principal component analysis is capable of reducing the problem's predictors combina-

tions to only two components, allowing the projection in a two-dimensional visualisation of similarities and differences between a set of observations, as well as the level of correlation between all predictors.

In this study, the Characterisation of the problem and Dimension reduction (Section 5.1) was developed around the methodology proposed by Neuenfeldt Júnior et al. (2017), maintaining all the characteristics observed in the original descriptive variables for the interpretation of how the problem characteristics affect the performance of the prediction.

Regression analysis based on data mining techniques provide an useful process for extracting information about the problem from a huge volume of data (Bastos, Lopes, & Pires, 2014; Perez et al., 2004). Modelling techniques can be used to describe the relationship between the characteristics and the predictions. The most common approaches are related with linear, logistic and polynomial regression models (Kumar & Vijayalakshmi, 2011). Stepwise, ridge, least absolute shrinkage and selection operator are robust techniques used mainly to solve situations with outliers, non-parametric data, or multicollinearity between variables. In contrast, they require more time to process all information (Brazdil, Soares, & Da Costa, 2003). To conduct the data mining Technique selection (Section 5.2), the comparison between the performance of different techniques can be evaluated using a statistical test of hypothesis (Demšar, 2006), comparing the coefficient of determination, R^2 , obtained with the use of different problem instances.

To develop the Regression analysis (Section 5.3), the definition of predictors and one known response variable are required. In this research, components obtained during the dimension reduction process are used as predictors. The response variable is obtained by the gap between the best random weight local search solution (using the bottom-left-fill as constructive heuristic) and the area lower bound. Finally, the problem instances in Dataset generated by the 2DCPackGen problem generator (Silva et al., 2014) is proposed to represent the problem. After the regression analysis, an Additional validation (Section 5.4) test must be conducted to validate the level of generalisation of the regression model and an absolute predictions' accuracy, using additional random problem instances and benchmark problem instance datasets (Validation dataset) to predict the strip height in different conditions.

To Assessing framework performance (Section 6), the Framework usage (Section 6.1) to predict the strip height for new problem instances is explored. Firstly, the characteristics of a new problem instance must be quantified according to the descriptive variables used to represent the 2D-SPP. Secondly, these quantified descriptive variables are converted into predictors during the dimension reduction process. Finally, the prediction's performance as stopping criterion to avoid long computational times is evaluated.

The Prediction performance analysis (Section 6.2) was developed to verify if small perturbations in the strip height predictions are capable of substantially changing the local search stopping process behaviour, considering the relation between the quality of the solutions, the number of iterations and variations in the predictions.

4. Problem knowledge

This section focuses on the introduction of what's required in order to use the knowledge discovery about the problem as input data for the regression analysis. Firstly, Section 4.1 describes the parameters and configurations to generate the problem instances of Dataset. Section 4.2 describes the bottom-left-fill heuristic concepts and the area lower bound, used in this research as a reference value for the problem instances. Both heuristic and area lower bound were used to define the response variable of the regression model.

4.1. Dataset

The use of data mining techniques to predict the strip height requires a large problem instance dataset, mainly to accurately represent the limits and behaviour of the problem, and also to consider the influence of aspects and characteristics that affect the quality measure of local search solution values (Bortfeldt, 2006).

Problem instances in the 2D-SPP are divided into two types: zero-waste and non-zero-waste. The zero-waste problem instances are generated by successively cutting a strip in smaller rectangles, and consequently the optimal solution does not allow waste space after positioning all the rectangles inside the strip. In non-zero-waste problem instances, the optimal solution is generally not known. The area lower bound value is almost equal to the optimal value when the optimal solution allows less waste space. An optimal solution with more wasted space has more uncertainty in the optimal arrangement of the rectangles in the strip, resulting in a less accurate area lower bound.

The problem instance generation is directly related with the problem generator's ability to combining rectangles and strip geometrical possibilities (Wang & Valenzuela, 2001). Based on a fixed input information, a greater variation of the characteristics demands the use of the problem generator's capability to correctly interpret the information available, returning as output a proportional number of problem instances.

In this study, the Dataset is composed by 30,000 problem instances created using the 2DCPackGen problem generator (Silva et al., 2014). The 2DCPackGen allows the generation of a large number of problem instances, using different parameters under controllable aspects, and ensures the reproducibility of the data. For the generation, information on a minimum and maximum value for a set of parameters is required.

Specifically in the 2D-SPP, the parameters that must be defined are: the rectangle minimum and maximum size dimension (1–100); the strip minimum and maximum width (10–1,000); the minimum and maximum number of different rectangle types (5–500); and the minimum and maximum number for the rectangle type demand (1–10). The problem instance variation is fitted using a beta probabilistic distribution (Gupta & Nadarajah, 2004). Different curve behaviours represent different geometric rectangles and strip shapes. The minimum and maximum values of each parameter were defined based on the most common values found in the benchmark problem instances of the literature.

To avoid the generation of similar problem instances, 10 different classes were developed, based on different characteristics of the size and shape of the rectangles.

In each class, the strip width, the number of different rectangle types and the rectangle type demand vary, respectively, according to two, five and three distribution curves, representing 30 different combinations, named as subtypes. For each subtype, 100 similar problem instances were generated, resulting in a total of 3000 problem instances in each class.

For example, the *pt7_28_9* is a class 7 problem instance (Fig. 3), where the probability of the rectangles being narrow (w_1) or long (w_2) and tall (h_1) is higher. In this research, we deal with the rotation condition, so the rectangle dimension is not fixed, which also allows obtaining in class 7 long (w_3) and short (h_4) or tall (h_3) rectangles. The subtype of this problem instance is 28, where the probability of the strip width being of an intermediate size is higher, different rectangle types are defined by an uniform distribution, and a large number of rectangles in each type is highly probable.

Most of the benchmark problem instances of the literature are very similar, which does not provide sufficient diversity to cover some important aspects of the strip-packing characteristics. Fig. 4 illustrates the behaviour of the 1270 most used benchmark prob-

lem instances from the literature and the 30,000 generated problem instances for Dataset, based on two basic metrics: for all rectangles, $r \in R$, the mean aspect ratio shape¹ between the maximum, $d1_r$, and minimum, $d2_r$, rectangle dimensions and the strip aspect ratio² between the area lower bound, L_0 , and the strip width W .

A considerable number of practical applications of the 2D-SPP involve the cutting of strip and rectangle geometry closest to the square shape. This trend is reflected in the way the benchmark problem instances were generated over the years, a fact verified by the density of points in the lowest part of Fig. 4a, where strips and rectangles are geometrically square, rather than narrow. Most of the generated problem instances also have this type of geometry. However, the objective of the regression analysis is to represent most of the possible behaviours of the problem through the generation of different problem instances. For example, a series of problem instances are formed by narrow strip and square rectangles (points at the top of Fig. 4b), by square strip and narrow rectangles (points at the right), or by narrow strip and rectangles (points at the upper right corner).

4.2. Technical analysis

In this study, the Bottom-Left-Fill (BLF) heuristic (Chazelle, 1983) was selected as constructive heuristic. The BLF is a positioning-based constructive heuristic that fits individual rectangles into the lowest left position free space identified in the strip. A complete solution is obtained when the last rectangle is packed into the strip.

The BLF is one of the most used constructive heuristics in the 2D-SPP literature, being efficient to solve both small and large problem instances. The BLF considers the free spaces between already packed rectangles as feasible positions, which is impossible in the Bottom-Left (BL) heuristic proposed by Baker, Coffman, and Rivest (1980). Thus, the BLF is much more complex ($O(n^3)$, where n is the number of rectangles) when compared with the BL ($O(n^2)$).

Originally, the rectangles in the BLF were sorted by non-increasing area (Riff, Bonnaire, & Neveu, 2009). With the aim of increasing the search solution space, Hopper and Turton (2001a) proposed the use of four different sequences (by rectangle height, width, perimeter, and area), and the solution with the smaller height is selected. Based on this idea, we developed a BLF heuristic with a local search containing 100 independent sequences, using different types of input sequences, named as random weight local search. In the first four sequences, the rectangles are ordered by decreasing area, perimeter, width, and height. The following 76 sequences relate to a random weighted order procedure, divided into four parts containing 19 sequences, based on area, perimeter, width, and height. Finally, the last 20 sequences are fully randomly generated.

In the random weight procedure, each rectangle has a probability of being chosen to occupy the first position of the sequence. Thus, the rectangles' ordering is not completely random. This probability relates to the rectangles' geometric characteristics (area, perimeter, width or height dimensions). At each time, a rectangle is added to the sequence, and the probabilities of the remaining rectangle to be positioned in the sequence are updated.

The value used as a reference to define which rectangle will be positioned is defined randomly, from 0 to 0.999. The rectangles' probability is summed until a random value is reached. When this sum is greater than the reference value, the rectangle chosen to be positioned is the last but one.

The objective of the 100 sequences is to have a wide search space and boost the quality of the solution, avoiding local optima

¹ $\sum_{r \in R} d1_r/d2_r$.

² L_0/W .

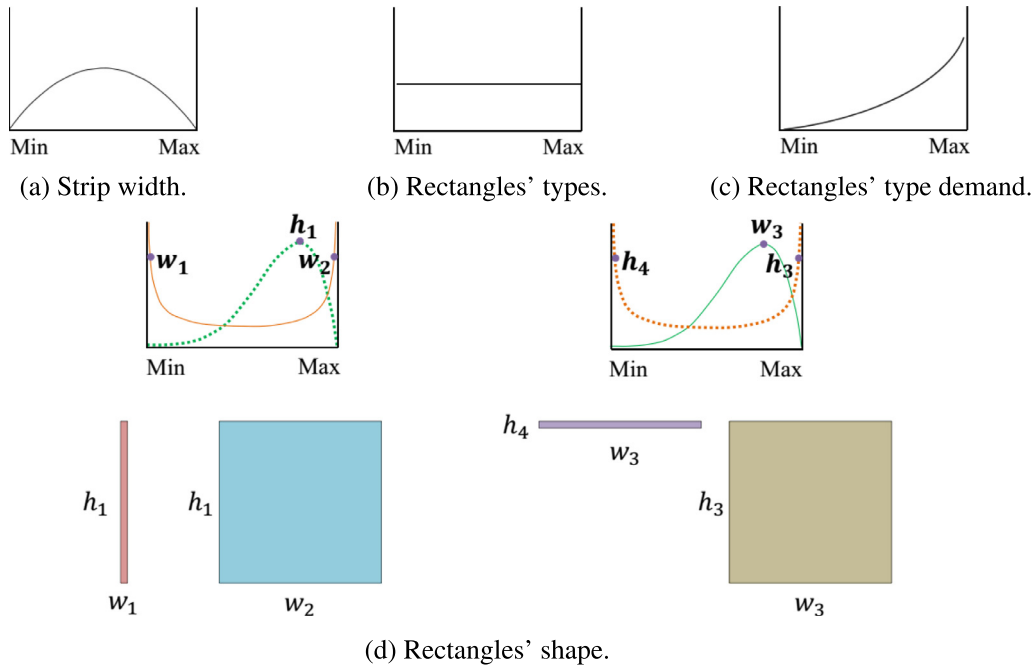


Fig. 3. Distribution curves used to generate the problem instance *pt7_28_9*.

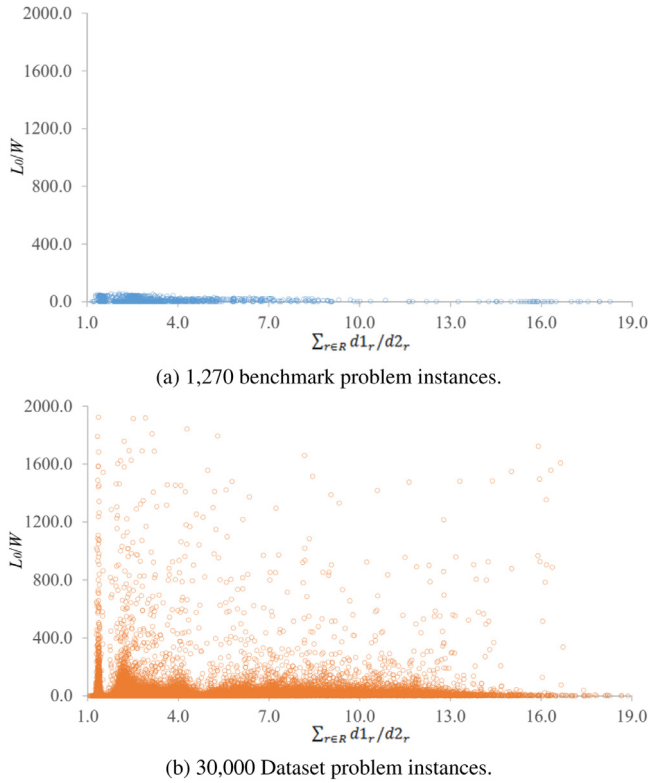


Fig. 4. Distribution of the problem instances ratio between strip aspect ratio shape (L_0/W) and all rectangles mean aspect ratio shape ($\sum_{r \in R} d1_r/d2_r$).

solutions. Finally, the best solution of the random weight local search is the one with the smallest height from all 100 solutions, and is defined as the reference strip height for the regression analysis, represented by H^{ref} . The best solution is used in the calculation of the response variable for the regression model.

The relaxation of the original problem in simple formats allows the calculation of the lower bounds. As an example, the 2D-SPP

can be relaxed in a one-dimensional contiguous bin-packing problem, successively cutting the rectangles' height (or width) for one unit, while maintaining the second dimension in the original format. In this case, the sum of the height of all bins is the solution for the new problem. While it is not feasible for the original 2D problem, it can be considered as a valid lower bound (Martello et al., 2003). Other concepts explored in the literature determine the geometrical differences between the rectangles and the strip width through linear combinations, using dual feasible functions, rectangle area or rectangle height (Bekrar & Kacem, 2009; Boschetti & Montaletti, 2010).

The continuous lower bound³ L_c is calculated using the total area of all rectangles R ($r \in R$) and the strip width, computing in a linear time. Considering $d1_r$ and $d2_r$, respectively, as the tallest and smallest dimensions of the rectangle, the maximum dimension $d2_r$ between all rectangles R is defined as lower bound L_h . The area lower bound, L_0 , is the maximum value found between L_c and L_h (Boschetti & Montaletti, 2010; Martello et al., 2003). Despite being a limited area lower bound, the continuous lower bound is the only one in the rectangular 2D-SPP that considers rectangle rotation. Therefore, this lower bound will be used as a reference value to calculate the response variable of the regression model.

With the values of the best solution found using the random weight local search, H^{ref} , and the area lower bound, the response variable can be described in a relative normalised format. The use of a single absolute variable, i.e. H^{ref} , is not recommended to answer the characteristics described by the predictors, because of its high variability even when problem instances have similar characteristics.

Thus, the use of area lower bound as a reference value (L_0) must be proposed to evaluate the quality of the solutions H^{ref} found. In our research, the $gap^{ref}(L_0)$ is the normalised value to be used as a known response variable in the regression analysis.

³ $L_c = (\sum_{r \in R} d1_r \cdot d2_r) / W$.

⁴ $gap^{ref}(L_0) = (H^{ref} - L_0) / L_0$.

5. Data mining approach

This section aims to incorporate the information about the problem in order to develop the regression analysis and predict the strip height. Firstly, the process of extracting characteristics of the problem and the dimension reduction using the principal component analysis are described in Section 5.1. A total of 19 predictors were selected to be used as explanatory variables in the regression analysis. The response variable is calculated by taking into consideration the $gap^{ref}(L_0)$ between the best solution of the random weight local search, H^{ref} , and the area lower bound L_0 .

A data mining selection step was conducted in Section 5.2 with the aim of selecting the technique that best fits the Dataset problem instances. In Section 5.3, the regression model was fitted. Finally, in Section 5.4, the level of generalisation (given by the coefficient of determination R^2) of the regression model and an absolute predictions' accuracy (given by the root-mean-square-error $RMSE$) were validated using a new set of 6000 problem instances generated using the 2DCPackGen (Silva et al., 2014) problem generator and the set of the most used non-zero-waste benchmark problem instances from the literature.

5.1. Problem characteristics and dimension reduction

The study of the most important 2D-SPP characteristics aims to find combinations between the rectangles and the strip size and shape characteristics (Hall & Posner, 2007; Smith-Miles & Lopes, 2012). The main idea was to transform all the characteristics into a limited number of quantitative measures, named as descriptive variables.

Since the number of descriptive variables can be very high, a Principal Component Analysis (PCA) was proposed in order to reduce the problem complexity and to facilitate a regression analysis interpretation. In addition, the reduction of descriptive variables in components helps to better understand the relation between the predictions for each problem instance and the problem characteristics.

PCA finds patterns in high dimension data, retaining most of the original information of each descriptive variable. The aim is to maximise the variance explained by each uncorrelated component extracted, boosting a wide representation of the different problem characteristics (López-Camacho & Terashima-Marín, 2013).

A recent work using PCA for the 2D-SPP considering the 1270 benchmark problem instances most used in the literature was proposed in Neuenfeldt Júnior et al. (2017). A total of 56 descriptive variables were defined and divided into six groups (*Area*, *Perimeter*, *Dimensions*, *Widthdimensions*, *Proportions*, and *Other*) according to the similarity of each descriptive variable within the group.

The descriptive variables were defined based on the parameters and concepts found in the problem generators that have been proposed over the years in the literature, not only for the 2D-SPP, but also for other cutting and packing problems (Beasley, 1985b; Berkey & Wang, 1987; Bortfeldt & Gehring, 2001; Ferreira & Oliveira, 2005; Hall & Posner, 2001; Leung, Zhang, & Sim, 2011; Silva et al., 2014; Wang & Valenzuela, 2001).

In this work, the methodology proposed in Neuenfeldt Júnior et al. (2017) will be used. The same 56 variables will be used to characterise the 30,000 problem instances generated for Dataset.

As suggested in Neuenfeldt Júnior et al. (2017), the PCA was individually applied to the *Area*, *Perimeter*, *Dimensions*, *Widthdimensions*, and *Proportions* groups, and two components with eigenvalues greater than or equal to one were extracted for each group, resulting in a total of 10 components. The division into five groups maintains the main characteristics of the descriptive variables and ensures a high mean variance explained for all groups. Table 2

Table 2

Variance explained by each component for all groups.

Group	Component	Eigenvalue	Variance (%)	Cumulative (%)
Area	<i>areacomp</i>	7.05	78.31	
	<i>areastats</i>	1.18	13.14	91.45
Perimeter	<i>perimcomp</i>	5.80	64.39	
	<i>perimstats</i>	2.25	24.94	89.33
Dimensions	<i>dimcomp</i>	6.60	66.03	
	<i>dimstats</i>	2.52	25.24	91.27
Width dimensions	<i>widthdimcomp</i>	6.24	69.33	
	<i>widthdimstats</i>	1.66	18.45	87.78
Proportions	<i>propcomp</i>	5.45	54.52	
	<i>propstats</i>	3.73	37.25	91.77

shows the components extracted by each group. Components *areastats*, *perimstats*, *dimstats*, *widthdimstats*, and *propstats* are based on variables characterized by classical statistical measures (mean, median and, standard deviation). In contrast, components *areacomp*, *perimcomp*, *dimcomp*, *widthdimcomp*, and *propcomp* are influenced by ratio (between percentiles or quartiles measures) and composition (between the sum of larger and smaller measures) variables.

Specifically, the group *Area* has the higher difference between the components' eigenvalues, while in group *Proportions*, the components have the most similar eigenvalues, being completely influenced by different descriptive variables. The group *Other* is atypical and does not present significant similarity between descriptive variables (small correlation coefficients). The PCA does not effectively reduce the group *Other* to a small number of components. Besides, descriptive variables with low correlations originate the extraction of components that are not intuitive and of difficult interpretation. The cumulative variance for the four components is only moderate (81.55%), and it is not comparable with the remaining groups. *Other* is composed by nine descriptive variables, based on the total number of items in the test problem instance (n), average items dimensions values (*coefficient*), proportion of different items (*heterog*), proportion of different items with more than one item (*heterognt*), the total number of different items dimensions (*difcoefficient*), the number of times that the object lower-bound is bigger than the object width (*objdimratio*), the number of times that the items the maximum items dimension is bigger than the minimum items dimensions (*itdimratio*), 10% larger items dimensions values (*maxcoefficient*), and 10% smaller items dimensions values (*mincoefficient*).

For factor loading projections, all groups present a similar behaviour. Composition and ratio descriptive variables have a more significant influence on the first components (*areacomp*, *perimcomp*, *dimcomp*, *widthdimcomp* and *propcomp*). Instead, most classical statistical descriptive variables (e.g. mean, median) have more influence on the second components (*areastats*, *perimstats*, *dimstats*, *widthdimstats* and *propstats*). The behaviour observed in the PCA extraction is not only dependent on the groups, but also on the nature of the descriptive variables. This fact is more visible in group *Proportions*, in which none of the descriptive variables have a high influence for both components at the same time. The standard deviation descriptive variable (e.g. *areastdev*) is a special case where the impact on groups *Area*, *Dimensions* and *Widthdimensions* is higher for the first components, and on groups *Perimeter* and *Widthdimensions*, the factor loading affects the second components.

Five descriptive variables (*areamean*, *areastdev*, *perimstdev*, *dimmean* and *widthdimmean*) have a complex structure, influencing both components in groups *Area*, *Perimeter*, *Dimensions* and *Widthdimensions*. As an example, the *perimstdev* factor loading is equal to 0.6 for *perimcomp* and 0.7 for *perimstats*. By means of variance, the *perimstdev* descriptive variable can accurately explain both components, even without a high factor loading. Furthermore,

perimstdev can influence a third component with an eigenvalue lower than one, which will not be extracted by the PCA. Finally, none of the groups has a non-influential descriptive variable for the extracted components. In at least one of the components, the factor loadings of the descriptive variables are always higher than or equal to 0.4.

For group *Other*, the descriptive variables' dispersion is high, and the components are explained by a small number of descriptive variables with high factor loading. This dispersion also shows the need for more components to provide a better representation of the group characteristics.

Four components were extracted with an eigenvalue higher than or equal to 1. If new components were added, the cumulative variance explained would increase. However, PCA would not significantly reduce the dimensions of the group *Other*.

Regarding the last analysis, the components proposed by the PCA to the group *Other* are not considered in our study. For the regression analysis, the 19 predictors are composed of ten components provided by the PCA applied in the groups of descriptive variables *Area*, *Perimeter*, *Dimensions*, *Widthdimensions* and *Proportions*, and directly the nine descriptive variables that defines the group *Other*.

5.2. Technique selection

The data mining technique that better adjusts the Dataset was selected taking into consideration three phases. Firstly, Friedman non-parametric tests were conducted based on Demšar (2006) procedures. Secondly, if the null-hypothesis was rejected, a post-test using the critical difference, *CD*, verified which technique could be distinguished from one another. Finally, in the group of non-significant distinguished techniques, the higher coefficient of determination, R^2 , was used to select the data mining technique.

In R^2 , the results for the $gap^{ref}(L_0)$ were compared with the predicted $\hat{gap}^{ref}(L_0)$, obtained using different data mining techniques. If the predictions have a good fit with the calculated $gap^{ref}(L_0)$, then the R^2 is high. In contrast, lower R^2 indicates that the $gap^{ref}(L_0)$ variation is not well explained by the predictors. This low level of adjustment can be explained by the lack of predictors capable of accurately explaining the phenomena, resulting in an incomplete representation of the main problem characteristics. Another reason is the intrinsic data mining technique incapacity of fitting a regression model capable of precisely predicting the $\hat{gap}^{ref}(L_0)$.

A total of 13 parametric and non-parametric techniques were previously selected for testing: two Random forest (*rf** and *rborist*); the Stochastic and Extreme gradient boosting (*gbm* and *xgbTree*); the Extreme learning machine (*elm*), the Back-propagation and Bayesian regularised neural networks (*nnet* and *brnn*); the Ridge regression (*ridge*); the Least absolute shrinkage and selection operator (*lasso*); the Linear and Multivariate adaptive regression models (*lm* and *mars*); the k-nearest neighbours (*kknn*); and the Cubist (*cub*). All techniques are capable of predicting the strip height based on supervised regression analysis concepts, tested using the software *RStudio*, specifically the function “train” in package “caret” (Kuhn, 2008). To avoid interfering in the results, all techniques were tested without the definition of any specific input parameter.

From the 30,000 Dataset problem instances, a small sample of 1000 problem instances was selected (100 of each class). To proceed with Friedman's hypothesis tests, this sample of 1000 problem instances was randomly divided into five subsamples, each with two different types of problem instances: training (800) and test (200). Regression models were fitted for each technique using the five training subsamples described, and Fig. 5a shows the mean

R^2 values from the difference between $gap^{ref}(L_0)$ and $\hat{gap}^{ref}(L_0)$ of test subsamples.

When the calculation of the performance of the techniques in each subsample shows a significance level equal to 0.05, the null-hypothesis is rejected, and there is a significant difference in the R^2 performance between the data mining techniques.

The Nemenyi (Demšar, 2006) post hoc test uses the critical difference, *CD*, to verify which techniques are significantly similar. Fig. 5b shows in the Nemenyi scale the mean positioning performance of all techniques compared with the calculated *CD*. Using as a reference value the technique with the best positioning performance (*cubist*), techniques are significantly equal if the position performance is located between 0 and 6.8.

The first six techniques with the highest R^2 (*rf**, *gbm*, *xgbTree*, *cub*, *kknn*, and *rborist*) do not have significant differences. Using the five subsamples adopted, none of these techniques stood out as having superior performance. As an alternative, the mean coefficient of determination was defined as an alternative selection measure. Therefore, with R^2 equal to 0.62, the Random Forest (RF) was selected to perform the height prediction, as it is the data mining technique with the best mean R^2 from all available options.

5.3. Regression analysis

The RF is a supervised and non-parametric ensemble data mining technique to fit classification or regression models, using series of individual random decision tree structures. Each decision tree has nodes and arcs. The nodes are labelled by input features (predictors), organised in levels. The arcs coming from the nodes are labelled by possible values obtained for the response variable, the $gap^{ref}(L_0)$.

The main objective is to select in each node the predictor that produces the prediction $\hat{gap}^{ref}(L_0)$ most similar to the calculated $gap^{ref}(L_0)$. For the regression model, the difference between the mean-square-error before and after the addition of a specific predictor in the node is the measure used to select the best predictor. The decision tree grows until a stopping condition is reached (e.g. maximum number of levels, or root-mean-square-error lower than or equal to a specific value). Each decision tree has the same weight, and the final RF regression model are defined by the mean predictions $\hat{gap}^{ref}(L_0)$ obtained by the construction of all individual random decision trees developed.

In this research, the RF regression analysis was fitted using a concept called “bagging”, to improve the regression model level of generalisation. The 30,000 Dataset problem instances were randomly divided into two parts: training (24,000) and testing (6,000). A total of 150 trees were considered, with all predictors evaluated at each split. The holdout method was applied to create trees with 66% of training problem instances (“bag” observations), defined in a random process. The remaining 33% of training problem instances excluded from this tree, “out-of-bag” (OOB) observations, were used to predict the mean-square-error between $gap^{ref}(L_0)$ and $\hat{gap}^{ref}(L_0)$ in each tree. The sampling with replacement option allows any problem instance to appear multiple times within the “bag” or “out-of-bag” groups.

To improve the quality of regression models, the 5-fold cross-validation process was conducted, folding the training sample into five different parts. The problem instances were randomly inserted into only one fold. A total of ten regression models were generated, two for each fold, and the regression model with the lowest residual-sum-of-squares-error between $gap^{ref}(L_0)$ and $\hat{gap}^{ref}(L_0)$ was selected as the RF regression model.

To verify the predictors' performance, we adopt the proportional *IncNodePurity* relevance index. The *IncNodePurity* measures the difference between the mean-square-error (for “out-of-bag

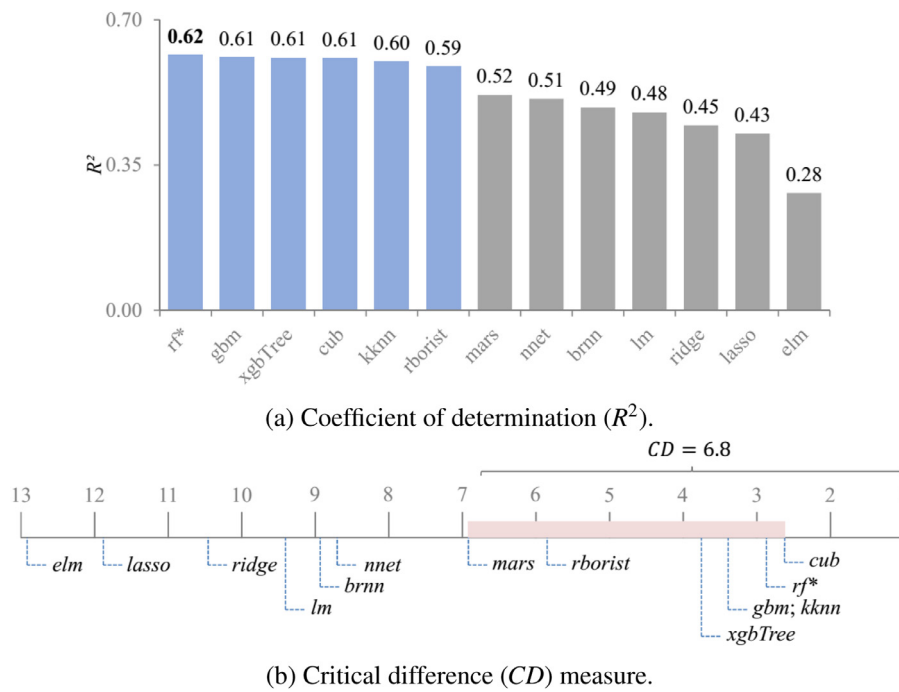
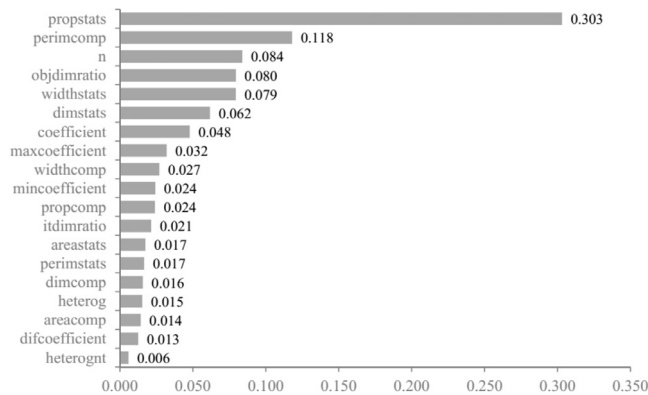


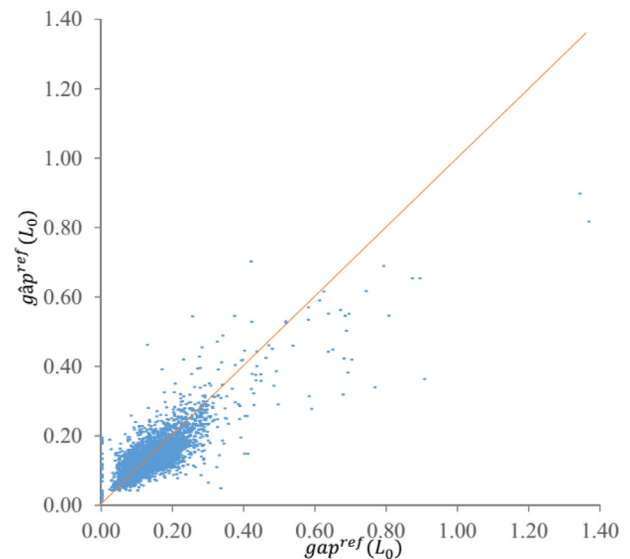
Fig. 5. For each data mining technique.

Fig. 6. The proportional *IncNodePurity* predictors' importance.

problem instances) between $\hat{gap}^{ref}(L_0)$ and $gap^{ref}(L_0)$ before and after the tree split (node). The most relevant predictors are more efficient in reducing the mean-square-error and have higher proportional *IncNodePurity* relevance values. The average value obtained for all RF trees is calculated for each predictor.

The results of the proportional *IncNodePurity* relevance index can vary from one RF regression model to another. Three main parameters affect the performance: (1) the number of trees; (2) the total number of predictors tested in each node; and (3) the random process of developing the decision trees. In this research, we fixed the first two parameters, the number of trees (150) and the total number of predictors tested in each node (19). For the third parameter, the random processes of developing the decision trees, the RF was run ten times, developing random and different trees for each RF. Fig. 6 shows the mean proportional value *IncNodePurity* for the 19 predictors after running the RF ten times.

The *propstats* and *objdimratio* generated the largest mean-square-error reductions before and after split the tree, which is sufficient to consider that these predictors explain more accurately most of the problem characteristics. In addition, the selection of

Fig. 7. Calculated $gap^{ref}(L_0)$ and predicted $\hat{gap}^{ref}(L_0)$ for the 6000 Dataset test problem instances.

propstats and *objdimratio* as the most important predictors is directly related to the manner in which the RF processes all input data to fit the regression model. In general, none of the six groups of predictors is predominant, with one predictor of each group included in the five highest proportional *IncNodePurity* values.

The regression model generated using the RF for the 24,000 Dataset training problem instances was applied to obtain the predictions. Fig. 7 shows the dispersion graph obtained for the calculated $gap^{ref}(L_0)$ and predicted $\hat{gap}^{ref}(L_0)$ response variable for the 6000 Dataset test problem instances. The coefficient of determination, $R^2 = 0.68$, and the predictions' accuracy $RMSE = 0.04$ are on a good level.

For almost all problem instances (5,865 out of 6000), the calculated $gap^{ref}(L_0)$ is in the range between 0.02 and 0.3. The regres-

sion model performs more accurate predictions ($RMSE = 0.03$) for problem instances located within this range. Conversely, for problem instances with calculated $gap^{ref}(L_0)$ higher than 0.3 (97 out of 6000) or lower than 0.02 (38 out of 6000), the predictions are not accurate ($RMSE = 0.16$).

An important characteristic of the problem with impact on the results obtained for the predictions' accuracy is the shape of the strip. Among the different formats, two are described in detail: the narrow strip and the square strip. The narrow strip problem instances have a strip width W that is much smaller than the calculated height H^{ref} , where the strip aspect ratio⁵ is equal to or higher than 100. This specific case was observed in 304 test problem instances, and the predictions' accuracy ($RMSE = 0.10$) are not well fitted for almost all problem instances. For the square strip instances in which the strip width dimension W is almost equal to the calculated height H^{ref} , the strip aspect ratio is between 0.8 and 1.2. Considering all 233 square strip problem instances, the predictions have a good fit quality ($RMSE = 0.02$). For example, the problem instance *pt11_21_39* has a narrow strip and has the worst prediction accuracy of all test problem instances ($RMSE = 0.54$). On the other hand, the problem instance *pt7_38_1* has a square strip and the prediction accuracy is almost equal to zero ($RMSE = 0.004$).

In the case of the ten problem instance classes, the predictions' behaviours are very similar, because most of the calculated $gap^{ref}(L_0)$ are located in a very dense space between $gap^{ref}(L_0) = 0.02$ and $gap^{ref}(L_0) = 0.3$. This fact is directly related with the problem instance generation, which was based on the combination of four parameters (the rectangle size, the strip width, the number of different rectangle types, and the rectangle type demand), producing similar problem instances in each class.

5.4. Additional validation

A completely different problem instance dataset (named as Validation dataset) was used in the regression model fitted in Section 5.3 with the goal of verifying the level of generalisation R^2 between $\hat{gap}^{ref}(L_0)$ and $gap^{ref}(L_0)$ and the predictions' accuracy $RMSE$.

A total of 6986 Validation dataset problem instances from two different datasets were used, in which 6000 problem instances were additionally generated using the 2DCPackGen problem generator (Silva et al., 2014). The uniform distribution was adopted for all problem generator parameters, allowing the generation of problem instances different from the 30,000 Dataset problem instances used to generate the regression model. This prevents a specific non-uniform distribution from interfering in the generation of different problem instances. The remaining 986 problem instances were obtained from non-zero-waste benchmark problem instances of the 2D-SPP, considering: 500 *bwmv* (Berkey & Wang, 1987), 360 *AH* (Bortfeldt, 2006), 72 *nice* and *path* (Wang & Valenzuela, 2001), 16 *zdf* (Leung & Zhang, 2011), 13 *gcut* (Beasley, 1985b), 12 *ngcut* (Beasley, 1985a), 10 *beng* (Bengtsson, 1982), and 3 *cgcut* (Christofides & Whitlock, 1977). In 254 benchmark problem instances, the rectangles and the strip dimensions exceed the maximum dimension parameters defined in the 30,000 Dataset problem instances generated to develop the regression model. To maintain the minimum and maximum parameter standards, the rectangle and strip values were divided by 10 or by 100.

The next step was found predictions for these new 6986 Validation dataset problem instances, to verify the level of adjustment of the regression model fitted using the Dataset. As expected, Fig. 8 shows that the level of generalisation ($R^2 = 0.55$) and the predic-

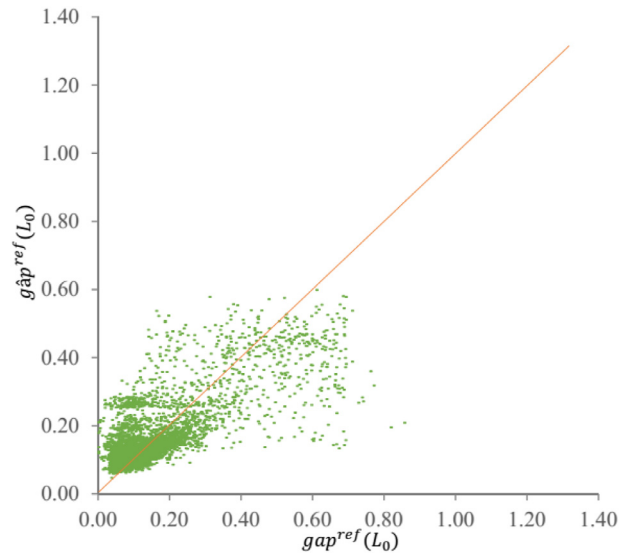


Fig. 8. Calculated $gap^{ref}(L_0)$ and predicted $\hat{gap}^{ref}(L_0)$ for the Validation dataset, composed by 6000 new problem instances generated and 986 non-zero-waste benchmark problem instances.

tions' accuracy ($RMSE = 0.08$) are lower than the results verified in the 6000 Dataset test problem instances, and the dispersion of results is higher when compared with the results obtained in Fig. 7.

These differences are due to the high degree of similarity between the test and training problem instances of the regression model, both generated simultaneously using same non-uniform distributions. Instead, benchmark problem instances were generated using different problem generators, each one with specific parameters and characteristics. For the 6000 problem instances additionally generated, the use of uniform distributions was proposed, thus differentiating their behaviour from Dataset.

As verified for the 6000 Dataset test problem instances, in the Validation dataset problem instances (6474 out of 6986) where the $gap^{ref}(L_0)$ is less than 0.3, the predictions are very accurate ($RMSE = 0.06$). This happens because of the high density of problems instances with similar $gap^{ref}(L_0)$ used to fit the regression model. For the Validation dataset problem instances (512 out of 6986) with calculated $gap^{ref}(L_0)$ higher than 0.3, the predictions' accuracy is not well adjusted ($RMSE = 0.19$).

6. Assessing framework performance

The main objective of this research is to propose a data mining-based framework to assess the quality of heuristic solutions for the 2D-SPP with 90° rotations. Predictions were obtained by a regression model fitted with the Random forest data mining technique, providing a quality measure to heuristic solutions and a stopping criterion to be used in local search algorithms in order to avoid long computational times.

To conduct the data mining approach presented in Section 5, a deeper knowledge about the problem was acquired through a 2D-SPP literature review (Neuenfeldt Júnior et al., 2017; Oliveira et al., 2016). After characterising the problem, a total of 19 predictors were defined, and one known response variable was calculated using the gap $gap^{ref}(L_0)$ between the lowest solution H^{ref} found using the random weight local search and the area lower bound L_0 .

Random forest was selected as best data mining technique option, based on the coefficient of determination, R^2 , measure and Friedman's hypothesis tests. The regression analysis was developed in order to find predictions $\hat{gap}^{ref}(L_0)$ of the response variable. Finally, the level of generalisation of the fitted regression model and

⁵ H^{ref}/W .

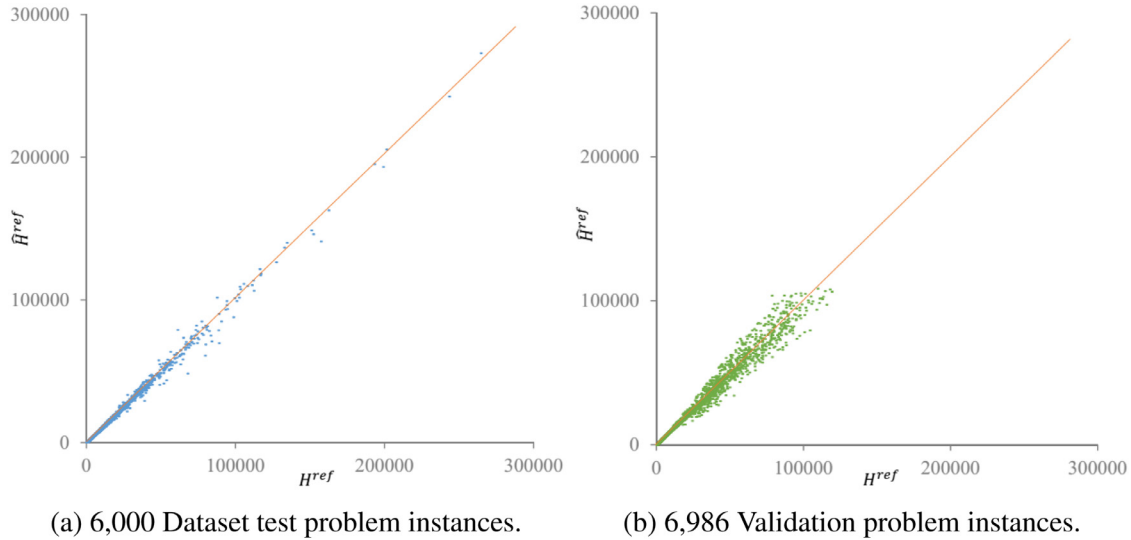


Fig. 9. Calculated H^{ref} and predicted \hat{H}^{ref} .

the predictions' accuracy were verified by calculating the predictions $\hat{gap}^{ref}(L_0)$ for 6000 Dataset test problem instances and 6000 Validation dataset problem instances.

To assess the framework performance, in this section the aim is to test the efficiency of the regression model fitted in order to predict the strip height in the 2D-SPP. Section 6.1 shows the conversion of $\hat{gap}^{ref}(L_0)$ in the strip height predictions⁶ \hat{H}^{ref} , and the use of \hat{H}^{ref} as a stopping criterion in the exchange procedure for three different local search algorithms (random weight, completely random, and dynamic random). In Section 6.2, a prediction performance analysis was conducted by varying the prediction value, \hat{H}^{ref} , in accordance with a multiplier γ in order to evaluate the robustness of the predictions according to the behaviour of the quality of the solutions of the dynamic random local search.

6.1. Framework usage

In this section, the use of the proposed data mining-based framework was proposed in two steps. Firstly, the regression model fitted in Section 5 is used to predict the strip height for the 6000 Dataset test problem instances used to predict the gap in Section 5.3 and 6986 Validation dataset problem instances are used to predict the gap in Section 5.4. Finally, the height predictions performance as stopping criterion was evaluated in three naive local search algorithms.

To test the efficiency of the regression model fitted in order to predict the strip height, Fig. 9 shows the results of the comparison between calculated H^{ref} and predicted \hat{H}^{ref} . For the 6000 Dataset test problem instances (Fig. 9a), almost all predictions \hat{H}^{ref} are very similar to the values H^{ref} obtained by the random weight local search. The level of generalisation is very high ($R^2 = 0.99$) and the measured predictions' accuracy is low ($RMSE = 955$). This high level of generalisation is closely related with L_0 , which is a fixed reference value used in both H^{ref} and \hat{H}^{ref} . In addition, for almost all 6000 Dataset test problem instances, $\hat{gap}^{ref}(L_0)$ and $gap^{ref}(L_0)$ have small RMSE values, resulting in small differences between the predicted strip height and the solution found using the random weight local search.

For the 6986 Validation dataset problem instances predictions (Fig. 9b), the level of generalisation ($R^2 = 0.98$) is also very high,

and the accuracy of the predictions can be considered of good quality ($RMSE = 2345$), mainly for the 2727 problem instances ($RMSE = 116$) with lower strip heights, where the calculated H^{ref} is less than 5000. The results of the regression model's lack of capability in accurately predicting the strip height for Validation dataset in comparison with Dataset can be seen mainly in problem instances with narrow strip, strips with high heights and strips with small widths. However, the adjustment obtained using the Validation dataset is acceptable and, as expected in this context, it does not significantly affect the good adjustment obtained for the predicted strip height \hat{H}^{ref} .

The regression model can be used to predict reference values for the local search algorithms exchange procedure, which can be used as the stopping criterion to avoid long computational times. Fig. 10 describes an evaluation of the predictions' performance as a stopping criterion for 1000 problem instances randomly provided by the Validation dataset (Section 5.4). The x-axis represents the 100 test iterations, and in the y-axis, the cumulative frequency containing the total number of problem instances that reached the predicted \hat{H}^{ref} is presented.

Three naive local search algorithms were adopted to improve the initial solution found with the BLF constructive heuristic with fixed sequences (by area, perimeter, width or height dimensions). Firstly, in the complete random local search, the sequence of rectangles is defined completely randomly and independently. Secondly, the random weight local search was used with the same characteristics defined to calculate the strip height of each problem instance for the $gap^{ref}(L_0)$ (response variable) of the regression model, as described in Section 4.2.

Finally, the dynamic random local search concepts were proposed, randomly changing the current rectangles' sequence position at each iteration by 5%. The initial solution is the best solution found that considers four fixed sequences, sorted by non-decreasing area, perimeter, maximum rectangle dimension, $d1_r$, and minimum rectangle dimension $d2_r$. The dynamic random local search explores the solution space applying local and small changes. The main advantage is maintaining significant information about the current best solution, which does not occur for the completely random and random weight local search algorithms, where the solution space is explored in a more general manner.

The adoption of more than one local search strategy was proposed in order to verify the behaviour of the predictions as stopping criterion for different situations. As expected, in all local

⁶ $\hat{H}^{ref} = L_0(1 + gap^{ref}(L_0))$.

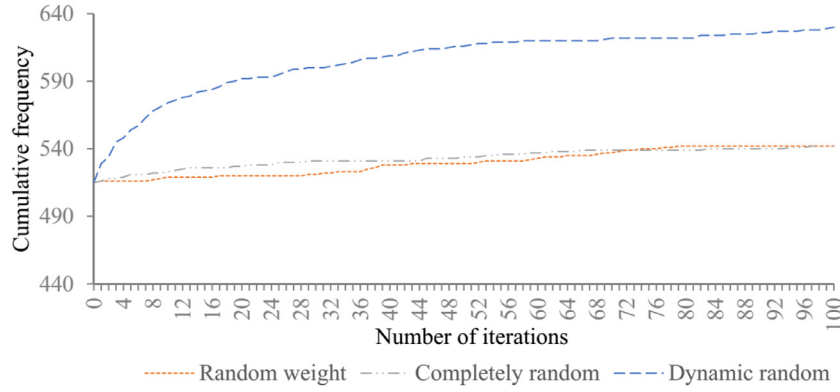


Fig. 10. Cumulative frequency of stopping process for the 1000 problem instances selected from Validation dataset.

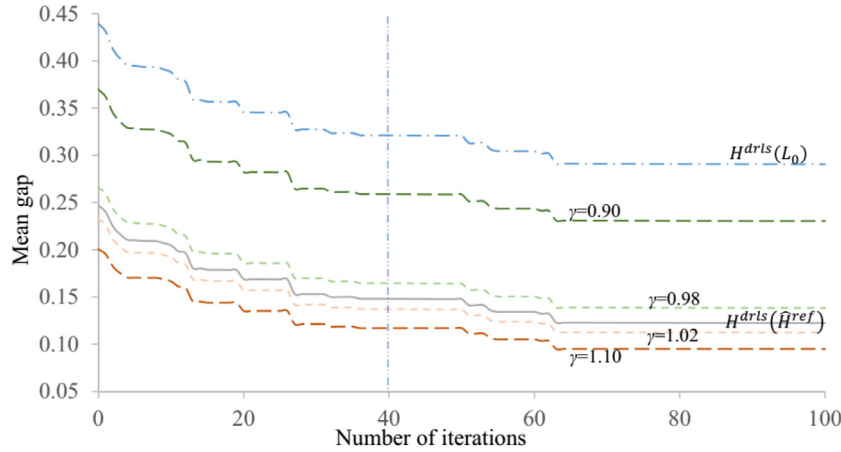


Fig. 11. Trade-off between the number of iterations and the quality of the solutions for the 1000 problem instances selected from Validation dataset, using different variations of the predictors and the area lower bound as stopping criterion. When the prediction \hat{H}^{ref} is adopted as a reference value for the local search, the gap is given by $gap^{dris}(\hat{H}^{ref})$. When the area lower bound is adopted as reference value, the gap is given by $gap^{dris}(L_0)$.

search algorithms, most of the problem instances reached the predicted \hat{H}^{ref} . The most effective criterion to stop the local search was to sort the rectangles by decreasing maximum and minimum dimensions. The dynamic random local search reached a higher number of heights, equal to or less than the predicted \hat{H}^{ref} , stopping a total of 630 problem instances out of 1,000, while the completely random and random weight local search algorithms have almost the same number of stops after 100 iterations.

The use of non-accurate predictions can have two different effects on the behaviour of the stopping criterion. Overly conservative predictions stop the local search early and the optimal solution is never reached. Conversely, overly optimistic predictions have smaller values than the calculated heuristic solutions, which may even be lower than the optimal solution. When using this criterion, the iterations will never stop.

6.2. Prediction performance analysis

The main objective of this section is to verify if small perturbations in the strip height predictions are capable of substantially changing the stopping process behaviour of the local search, considering the relation between the quality of the solutions obtained with the dynamic random local search, the number of iterations and variations in the predictions.

The predictions were not calculated by parametric models, which do not allow the use of metrics as the mean-square-error of the standard deviation as a means of verifying the predictions' robustness. The solutions obtained by the dynamic random lo-

cal search, using the \hat{H}^{ref} as stopping criterion, are represented by $H^{dris}(\hat{H}^{ref})$. The quality of these solutions is measured by the gap $gap^{dris}(\hat{H}^{ref})$ between the predictions \hat{H}^{ref} and the solutions $H^{dris}(\hat{H}^{ref})$. To evaluate the robustness of the predictions, small changes in the predictions were defined. For each problem instance, a γ value is multiplied by the prediction \hat{H}^{ref} and is used as the reference value to calculate the quality of the solution, $gap^{dris}(\hat{H}^{ref})$, provided by the dynamic random local search.

Fig. 11 shows the relation between the increase in the number of iterations (x-axis) and the reduction of the mean gap (y-axis) during the dynamic random local search algorithm, for the 1000 problem instances considered in Section 6.1, with the four different γ values adopted ($\gamma = 0.90$, $\gamma = 0.98$, $\gamma = 1.02$, and $\gamma = 1.10$). The study of the quality of the solutions $H^{dris}(\hat{H}^{ref})$ found using the dynamic random local search and the area lower bound as stopping criterion is also proposed in Fig. 11. It is measured using the gap⁷ $gap^{dris}(\hat{H}^{ref})$.

All curves present a similar behaviour, independently of the γ value considered. For $\gamma = 0.90$ and $\gamma = 1.10$, a variation of 10% in the original prediction \hat{H}^{ref} is considered, and, as expected, this has a high impact on the mean $gap^{dris}(\hat{H}^{ref})$, because with a high γ the local search reached the stopping criterion too early and a low γ presents optimistic predictions. Therefore, in the robustness analysis, $\gamma = 0.90$ and $\gamma = 1.10$ are not considered.

For $\gamma = 0.98$ and $\gamma = 1.02$, the mean $gap^{dris}(\hat{H}^{ref})$ is very similar for almost all problem instances during the local search. This

⁷ $gap^{dris}(\hat{H}^{ref}) = (H^{dris}(\hat{H}^{ref}) - \hat{H}^{ref}) / \hat{H}^{ref}$.

means that the quality of the solutions obtained is not greatly affected by small variations of the predicted \hat{H}^{ref} , stressing that the predictions were robust.

The graphic in Fig. 11 is also relevant to analyse how the reduction in the number of iterations affects the quality of the solution. Specifically, the mean gap $gap^{dris}(\hat{H}^{ref})$ considering prediction \hat{H}^{ref} is 12% after 100 iterations. If the number of iterations is limited to 40, the mean gap $gap^{dris}(\hat{H}^{ref})$ would be 15%, which means that saving 60 iterations would have an impact of only 3% in the quality of the solution. Also, the graphic shows the inefficiency of L_0 when used as a stopping criterion, demonstrated by the high values found for the gap⁸ $gap^{dris}(L_0)$ in comparison with the results obtained for all variations of \hat{H}^{ref} predictions. The L_0 is too conservative for almost all problem instances explored, thus requiring high computational processing times to provide the quality of the solutions $H^{dris}(L_0)$ found by the local search.

7. Conclusions

This research proposes a data mining-based framework capable of predicting a reference value to be used as stopping criterion in local search algorithms for the 2D-SPP, taking into consideration the main problem characteristics. The total height necessary to pack a set of small rectangular rectangles into a rectangular strip is the reference value used in the 2D-SPP. In addition, the predictions can provide a measure to verify the quality of solutions found by heuristics.

Data mining techniques were tested and the Random forest was statistically inferred as the best choice for developing the regression analysis, based on its ability to generalise the predictors for the normalized gap $gap^{ref}(L_0)$. Other data mining techniques, such as the Extreme gradient boosting, Cubist and k-nearest neighbours, also have significant generalisation ability. The input data was composed of three parts: the predictors to provide a numeric measure of the problem characteristics, the gap calculated using the area lower bound and the random weight local search, and 30,000 Dataset problem instances generated with 2DCPackGen (Silva et al., 2014) to represent different 2D-SPP variations found in real-world applications.

The 5-fold cross-validation in Section 5.3 was adopted to verify if the regression model has a good level of generalisation and if it accurately predicts the strip height for new problem instances. An additional validation proposed in Section 5.4 with the use of 6000 problem instances was generated together with 986 non-zero-waste benchmark problem instances. Both 5-fold cross-validation and additional validation confirmed the good level of generalization of the regression model and the predictions' accuracy. The quality of the level of adjustment obtained was fundamental to conclude that the extraction of problem characteristics was well performed.

In the assessing performance section, it was verified that the strip height predictions are highly influenced by the solution obtained by the constructive heuristic, by local search algorithms and by the reference values (area lower bound or the predictions found with the regression model fitted) used to calculate the gap. Therefore, the behaviour of the quality of the solutions was proved to be robust even when small changes in the predictions are applied. In addition, it was confirmed that the area lower bound is an overly conservative stopping criterion for almost all 1000 problem instances from Validation dataset used to test the predictions' robustness.

Widely used in literature over the years, the BLF was selected as the constructive heuristic due to the good cost-benefit ratio be-

tween its simplicity of implementation and the quality of the solutions found. Other constructive heuristics could have been used instead of BLF.

This research contributes to the identification of the characteristics that most affect constructive heuristics in solving the 2D-SPP. A better understanding of these characteristics brings new elements capable of shedding light on the design of new heuristics for the 2D-SPP, mainly when the optimal solution is unknown, reducing the computational time required to verify the quality of the solution. Furthermore, this research was used to validate the dimension reduction methodology proposed in Neuenfeldt Júnior et al. (2017).

In this study, we demonstrate that the data mining-based framework proposed is consistent and can be applied to predict response variable values in other problems.

Nevertheless, the work has natural limitations and weaknesses. The very first limitation, when using H^{ref} as a stopping criterion in search algorithms, is that it is not a bound but a reference value. This means that the value of the (unknown) optimal solution may be above or below H_{ref} . Being below the optimum may cause a worthless long number of iterations trying to improve what cannot be improved, and being above the optimum may cause a premature halting of the search. In fact, how to use a reference value instead of a bound, has still a large room for discussion and innovative ideas. Some directions for this innovative use of H_{ref} are provided in Section 6.2 when H_{ref} is multiplied by a factor γ , which takes values around 1. Being H_{ref} just a reference value, (fast) heuristics may be run several times with different stopping criteria, based on a percentage of H_{ref} . Additional, H_{ref} can be used in conjunction with true bounds already available in the literature. This strategy would mitigate (not eliminate) many of the drawbacks previously presented.

In the best theoretical case in which the regression model worked perfectly, the proposed methodology would give a good reference value for the solution obtained by a particular heuristic algorithm. However, the relationship between the value of the solution provided by this algorithm and the optimal solution of a given instance is not known, as the 2D-SPP is an NP-hard combinatorial optimization problem and guaranteed optimal solutions can only be achieved for toy problems. Our goal was to provide a reference value for the solution (and we can not call it a bound because we cannot be sure if the optimal solution is below or above this value) better than the existing trivial bound based on the rectangles' area. We really do not know whether the distance from the solution of this algorithm to the (unknown) optimal solution is similar for all types of instances or not, so to be near to the predicted value does not mean to be near to the optimum. Even more, while the optimal solution is only dependent on the data associated with each problem instance, our study has exclusively focused on a particular constructive heuristic (bottom-left-fill), and we just claim result validity for this particular heuristic. But, to the best of our knowledge, this is the first trial to predict the solution value in a cutting or packing problem by means of regression analysis, and we believe that this work may originate relevant future research.

For future research, it is needed to verify the level of adjustment of the predictions obtained by the regression analysis in other contexts. Specifically, in the 2D-SPP it is important to apply the predictions using the regression model as a stopping criterion in different local search algorithms. For each problem instance used to fit the regression model, a replacement process, similar to the backforward step of networks to adjust the response variable value, replacing the area lower bound by the predictions obtained after fit the regression model. In what concerns the use of supervised machine learning techniques, emerged as particularly interesting the exploration, in more detail, of neural network tech-

⁸ $gap^{dris}(L_0) = (H^{dris}(L_0) - L_0)/L_0$.

niques, increasing the number of observations in training dataset to improve the neurons' learning capacity to find patterns. Finally, the data mining based framework developed for the 2D-SPP can be extended to fit regression models for other types of cutting and packing problems, as the bin packing problem, knapsack problem cutting, and cutting stock problems.

Acknowledgements

The first author was supported by BE MUNDUS project awarded by the European Commission Erasmus Mundus programme within the grant [BM13DF0018/2014](#), and by the Brazilian National Council for Scientific and Technological Development within the grant BEX200936/2014.

The second author was supported by FCT – Fundação para a Ciência e a Tecnologia within the grant SFRH/BPD/98981/2013.

The work was also partially financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT – Fundação para a Ciência e a Tecnologia within project POCI-01-0145-FEDER-029609.

Appendix A. Notation

- r : rectangles' index;
- n : number of rectangles in the problem instance;
- $d1_r$: maximum rectangle dimension (for 2D-SPP with 90° rotations);
- $d2_r$: minimum rectangle dimension (for 2D-SPP with 90° rotations);
- w_1, w_2 , or w_3 : rectangle width dimension (for 2D-SPP without 90° rotations);
- h_1, h_3 , or h_4 : rectangle height dimension (for 2D-SPP without 90° rotations);
- W : strip width;
- OS : optimal solution;
- L_h : height lower bound, measured by the maximum dimension $d2_r$ between all rectangles for the problem instance;
- L_c : continuous lower bound⁹;
- L_0 : area lower bound;
- H^{ref} : random weight local search strip height lowest solution;
- \hat{H}^{ref} : predicted strip height related with the area lower bound L_0 ;
- $H^{drls}(\hat{H}^{ref})$: dynamic random local search strip height solutions based on the predicted \hat{H}^{ref} ;
- $H^{drls}(L_0)$: dynamic random local search strip height solutions based on the area lower bound L_0 ;
- $gap^{ref}(L_0)$: gap¹⁰ between the solution H^{ref} and the area lower bound L_0 ;
- $\hat{gap}^{ref}(L_0)$: predicted $gap^{ref}(L_0)$ related with the area lower bound L_0 ;
- $gap^{drls}(\hat{H}^{ref})$: gap¹¹ between the solution $H^{drls}(\hat{H}^{ref})$ and the predicted \hat{H}^{ref} ;
- $gap^{drls}(L_0)$: gap¹² between the solution $H^{drls}(\hat{H}^{ref})$ and the area lower bound L_0 ;
- γ : multiplier to vary the predictions \hat{H}^{ref} (for the robustness verification);
- CD : critical difference to distinguish the data mining techniques (for technique selection);

⁹ $L_c = (\sum_{r \in R} d1_r \cdot d2_r) / W$.

¹⁰ $gap^{ref}(L_0) = (H^{ref} - L_0) / L_0$.

¹¹ $gap^{drls}(\hat{H}^{ref}) = (H^{drls}(\hat{H}^{ref}) - \hat{H}^{ref}) / \hat{H}^{ref}$.

¹² $gap^{drls}(L_0) = (H^{drls}(L_0) - L_0) / L_0$.

- R^2 : level of generalisation given by the coefficient of determination (for data mining approach);
- $RMSE$: predictions' accuracy given by the root-mean-square-error (for data mining approach).

References

- Almeida, R. D., & Steiner, M. T. A. (2016). Resolution of one-dimensional bin packing problems using augmented neural networks and minimum bin slack. *International Journal of Innovative Computing and Applications*, 7(4), 214–224.
- Alvarez-Valdes, R., Parreno, F., & Tamarit, J. (2009). A branch and bound algorithm for the strip packing problem. *OR Spectrum*, 31(2), 431–459.
- Alvarez-Valdés, R., Parreño, F., & Tamarit, J. M. (2008). Reactive GRASP for the strip-packing problem. *Computers & Operations Research*, 35(4), 1065–1083.
- Baker, B. S., Coffman Jr, E. G., & Rivest, R. L. (1980). Orthogonal packings in two dimensions. *SIAM Journal on Computing*, 9(4), 846–855.
- Bastos, P., Lopes, I., & Pires, L. (2014). Application of data mining in a maintenance system for failure prediction. *Safety, Reliability and Risk Analysis: Beyond the Horizon*, 1(1), 933–940.
- Beasley, J. (1985a). Algorithms for unconstrained two-dimensional guillotine cutting. *Journal of the Operational Research Society*, 36(4), 297–306.
- Beasley, J. E. (1985b). An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, 33(1), 49–64.
- Bekrar, A., & Kacem, I. (2009). An exact method for the 2D guillotine strip packing problem. *Advances in Operations Research*, 2009(1).
- Bengtsson, B.-E. (1982). Packing rectangular pieces: A heuristic approach. *The Computer Journal*, 25(3), 353–357.
- Berkey, J. O., & Wang, P. Y. (1987). Two-dimensional finite bin-packing algorithms. *Journal of the Operational Research Society*, 38(5), 423–429.
- Bortfeldt, A. (2006). A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. *European Journal of Operational Research*, 172(3), 814–837.
- Bortfeldt, A., & Gehring, H. (2001). A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research*, 131(1), 143–161.
- Boschetti, M. A., & Montaletti, L. (2010). An exact algorithm for the two-dimensional strip-packing problem. *Operations Research*, 58(6), 1774–1791.
- Brazdil, P. B., Soares, C., & Da Costa, J. P. (2003). Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, 50(3), 251–277.
- Cadima, J., Cerdeira, J. O., & Minhoto, M. (2004). Computational aspects of algorithms for variable selection in the context of principal components. *Computational Statistics & Data Analysis*, 47(2), 225–236.
- Chazelle, B. (1983). The bottom-left bin-packing heuristic: An efficient implementation. *IEEE Transactions on Computers*, 100(8), 697–707.
- Christofides, N., & Whitlock, C. (1977). An algorithm for two-dimensional cutting problems. *Operations Research*, 25(1), 30–44.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1), 1–30.
- Fekete, S. P., & Schepers, J. (2001). New classes of fast lower bounds for bin packing problems. *Mathematical Programming*, 91(1), 11–31.
- Ferreira, E. P., & Oliveira, J. F. (2005). A note on Fekete and Schepers' algorithm for the non-guillotinable two-dimensional packing problem. *Technical Report*. FEUP.
- Gupta, A. K., & Nadarajah, S. (2004). *Handbook of beta distribution and its applications*. CRC Press.
- Hall, N. G., & Posner, M. E. (2001). Generating experimental data for computational testing with machine scheduling applications. *Operations Research*, 49(6), 854–865.
- Hall, N. G., & Posner, M. E. (2007). Performance prediction and preselection for optimization and heuristic solution procedures. *Operations Research*, 55(4), 703–716.
- He, Y., Wu, Y., & De Souza, R. (2012). A global search framework for practical three-dimensional packing with variable carton orientations. *Computers and Operations Research*, 39(10), 2395–2414.
- Hopper, E., & Turton, B. C. H. (2001a). An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. *European Journal of Operational Research*, 128(1), 34–57.
- Hopper, E., & Turton, B. C. H. (2001b). A review of the application of meta-heuristic algorithms to 2d strip packing problems. *Artificial Intelligence Review*, 16(4), 257–300.
- Kuhn, M. (2008). Caret package. *Journal of Statistical Software*, 28(5), 1–26.
- Kumar, S. A., & Vijayalakshmi, M. (2011). Efficiency of decision trees in predicting student's academic performance. In *First international conference on computer science, engineering and applications: Vol. 2* (pp. 335–343).
- Leung, S. C. H., & Zhang, D. (2011). A fast layer-based heuristic for non-guillotine strip packing. *Expert Systems with Applications*, 38(10), 13032–13042.
- Leung, S. C. H., Zhang, D., & Sim, K. M. (2011). A two-stage intelligent search algorithm for the two-dimensional strip packing problem. *European Journal of Operational Research*, 215(1), 57–69.
- Lodi, A., Martello, S., & Monaci, M. (2002). Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2), 241–252.
- López-Camacho, E., & Terashima-Marín, H. (2013). Evolving feature selection for characterizing and solving the 1D and 2D bin packing problem. In *IEEE congress on evolutionary computation (CEC)* (pp. 2094–2101). IEEE.
- López-Camacho, E., Terashima-Marín, H., Ochoa, G., & Conant-Pablos, S. E. (2013). Understanding the structure of bin packing problems through principal component analysis. *International Journal of Production Economics*, 145(2), 488–499.

- López-Camacho, E., Terashima-Marín, H., & Ross, P. (2010). Defining a problem-state representation with data mining within a hyper-heuristic model which solves 2D irregular bin packing problems. In *Ibero-American conference on artificial intelligence* (pp. 204–213). Springer.
- López-Camacho, E., Terashima-Marín, H., Ross, P., & Ochoa, G. (2014). A unified hyper-heuristic framework for solving bin packing problems. *Expert Systems with Applications*, 41(15), 6876–6889.
- Martello, S., Monaci, M., & Vigo, D. (2003). An exact approach to the strip-packing problem. *INFORMS Journal on Computing*, 15(3), 310–319.
- Neuenfeldt Júnior, A., Silva, E., Miguel Gomes, A., & Oliveira, J. F. (2017). The two-dimensional strip packing problem: What matters? *Congress of APDIO, the Portuguese Operational Research Society*, 223(1), 151–164.
- Neveu, B., Trombettoni, G., Araya, I., & Riff, M. C. (2008). A strip packing solving method using an incremental move based on maximal holes. *International Journal on Artificial Intelligence Tools*, 17(5), 881–901.
- Ntene, N., & van Vuuren, J. H. (2009). A survey and comparison of guillotine heuristics for the 2D oriented offline strip packing problem. *Discrete Optimization*, 6(2), 174–188.
- Oliveira, J. F., Neuenfeldt Júnior, A., Silva, E., & Carravilla, M. A. (2016). A survey on heuristics for the two-dimensional rectangular strip packing problem. *Pesquisa Operacional*, 36(2), 197–226.
- Özcan, E., Kai, Z., & Drake, J. H. (2013). Bidirectional best-fit heuristic considering compound placement for two dimensional orthogonal rectangular strip packing. *Expert Systems with Applications*, 40(10), 4035–4043.
- Perez, J., Frausto, J., Cruz, L., Fraire, H., & Santiago, E. (2004). A machine learning approach for modeling algorithm performance predictors. In *International conference on modeling decisions for artificial intelligence* (pp. 70–80). Springer.
- Rice, J. R. (1976). The algorithm selection problem. *Advances in Computers*, 15(1), 65–118.
- Riff, M. C., Bonnaire, X., & Neveu, B. (2009). A revision of recent approaches for two-dimensional strip-packing problems. *Engineering Applications of Artificial Intelligence*, 22(4–5), 833–837.
- Santoyo, A. M., Ortega, J. P., Vargas, D. R., & Reyes, L. C. (2015). Towards a characterization of difficult instances of the bin packing problem. *IEEE Latin America Transactions*, 13(7), 2454–2462.
- Silva, E., Oliveira, J. F., & Wäscher, G. (2014). 2DCPackGen: A problem generator for two-dimensional rectangular cutting and packing problems. *European Journal of Operational Research*, 237(3), 846–856.
- Smith-Miles, K., Baatar, D., Wreford, B., & Lewis, R. (2014). Towards objective measures of algorithm performance across instance space. *Computers & Operations Research*, 45(1), 12–24.
- Smith-Miles, K., & Lopes, L. (2012). Measuring instance difficulty for combinatorial optimization problems. *Computers & Operations Research*, 39(5), 875–889.
- Wang, P. Y., & Valenzela, C. L. (2001). Data set generation for rectangular placement problems. *European Journal of Operational Research*, 134(2), 378–391.
- Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3), 1109–1130.
- Wei, L., Zhang, D., & Chen, Q. (2009). A least wasted first heuristic algorithm for the rectangular packing problem. *Computers and Operations Research*, 36(5), 1608–1614.