Management and Production Engineering Review
Volume 12 • Number 4 • December 2021 • pp. 27–37
DOI: 10.24425/mper.2021.139992

A Multi-label Transformation Framework for the Rectangular 2D Strip-Packing Problem

Alvaro NEUENFELDT JÚNIOR, Matheus FRANCESCATTO, Gabriel STIELER, David DISCONZI

Federal University of Santa Maria, Production Engineering post-graduation program, Brazil

Received: 20 August 2020
Accepted: 03 November 2021

Abstract
The present paper describes a methodological framework developed to select a multi-label dataset transformation method in the context of supervised machine learning techniques. We explore the rectangular 2D strip-packing problem (2D-SPP), widely applied in industrial processes to cut sheet metals and paper rolls, where high-quality solutions can be found for more than one improvement heuristic, generating instances with multi-label behavior. To obtain single-label datasets, a total of five multi-label transformation methods are explored. 1000 instances were generated to represent different 2D-SPP variations found in real-world applications, labels for each instance represented by improvement heuristics were calculated, along with 19 predictors provided by problem characteristics. Finally, classification models were fitted to verify the accuracy of each multi-label transformation method. For the 2D-SPP, the single-label obtained using the exclusion method fit more accurate classification models compared to the other four multi-label transformation methods adopted.

Keywords
Strip-packing problem; Data mining; Multi-label transformation; Classification analysis; Heuristics.

Introduction

The rectangular 2D strip-packing problem (2D-SPP) is one of the most applied cutting and packing problems in the industry, as seen in the cutting of sheet metal and paper rolls. Given a set of small rectangles and a strip, the objective is to minimize the space to cut or position all rectangles into the strip, reducing possible waste. For the 2D problem, one of the dimensions of the strip is considered fixed, while the other is flexible. All rectangles must be into the strip without overlapping (Wäscher et al., 2007).

The 2D-SPP is an NP-hard problem and can be solved by exact methods and heuristics. In the exact methods, an optimal solution can be found and the optimality is proved. On the other hand, heuristics produce very good results with relatively low computational times, especially when the number of rectangles is very large, but the optimality cannot be proven (Alvarez-Valdés et al., 2008; Hopper & Turton, 2001; Martello et al., 2003; Ntene & Vuuren, 2009; Oliveira et al., 2016).

The heuristics are divided in constructive and improvement heuristics. In constructive heuristics, the rectangles are positioned into the strip according to determined criteria, such as bottom-left and best-fit, until obtaining a complete solution. In the improvement heuristics, a complete solution obtained by using any constructive heuristic is improved through successive modifications in the arrangement of the rectangles already positioned into the strip or by modifying the rectangles sequence ordering. For both, the first complete solution is improved with small consecutive changes until a stop criterion is reached (Oliveira et al., 2016).

Since the 80s, more than 30 improvement heuristics were developed to solve the rectangular 2D-SPP (Oliveira et al. 2016), which reveals the difficulty to select a good improvement heuristic option according to the industrial application represented by problem instances. Therefore, strategies to reduce the time required to select an improvement heuristic for the rectangular 2D-SPP must be adopted. Computer algorithms can be used to develop an efficient selec-

Corresponding author: *Alvaro Neuenfeldt Júnior – Federal University of Santa Maria, Avenue Roraima 1000 Santa Maria, Brazil, 97105-900 phone: (+55) 55-99192-7755, e-mail: alvaro.junior@ufsm.br*

tion process, as verified for categorical and numerical classification models based on supervised machine learning (SML) techniques (Glover, 1986; Brazdil et al., 2008).

Classification problems can assume the format of single-label and multi-label, depending on the intrinsic characteristics and context of the problem. The single-label format is related to problems where each instance is associated with a single solution, named as "label" (e.g. LA, LB, LC, ...). In the multi-label format, each instance can be associated with different solutions (Tsoumakas & Katakis, 2007; Horvath & Vircikova, 2012; Rogalewicz & Sika, 2016), as verified primarily for text categorization and medical diagnoses. As an example, in medical diagnoses, a patient may be suffering from more than one disease in the same period, such as hepatitis and flu. Modern applications are related to industrial optimization problems, as verified for the rectangular 2D-SPP, in which a quality solution can be found using more than one improvement heuristic. In quality solutions, the waste to position all the rectangles into the strip is reduced. Thus, the 2D-SPP can be characterized as a multi-label problem, a fact that prevents fit classification models using traditional SML techniques, requiring a method to transform a dataset with multi-label instances in single-label instances.

Given the difficulty to select good improvement heuristics options for rectangular 2D-SPP instances, the objective of this article is to describe a methodological framework developed to select a multi-label dataset transformation method. The main contributions of this article are as follows: a) Identify the best multi-label transformation method for the 2D-SPP, enabling fit classification models with acceptable levels of accuracy and generalization, based on single-label datasets consistent with the phenomena and characteristics observed for the 2D-SPP; b) Develop an adaptive methodological framework that can be used in other combinatorial optimization problems, mainly cutting and packing problems, as bin packing and knapsack problems; and c) From an academic perspective, filling a research gap related to the lack of studies applying a multi-label transformation method in the context of cutting and packing problems, in specific, the strip packing problem. The research gap was verified with a literature review on Web of Science and Scopus platforms, using as inclusion criteria for the research protocol: English language articles from scientific journals; publications before 2020; primarily keywords "strip", "packing", "open", "dimension", and "problem"; secondary keywords "rectangular" and "two dimensional"; and abstracts and titles related with the context.

The article was structured according to the following descriptions. Section 1 introduces the topic addressed in this paper. Section 2 shows the methodological framework. In Section 3, a contextualization about multi-label transformation methods was organized. In Section 4, a practical application of the methodological framework was proposed. Finally, Section 5 shows some of the main conclusions obtained during the research.

Methodological framework

This section aims to describe the methodological framework proposed to transform the multi-label datasets (Fig. 1). Concepts about each step of the methodological framework are presented in this section.

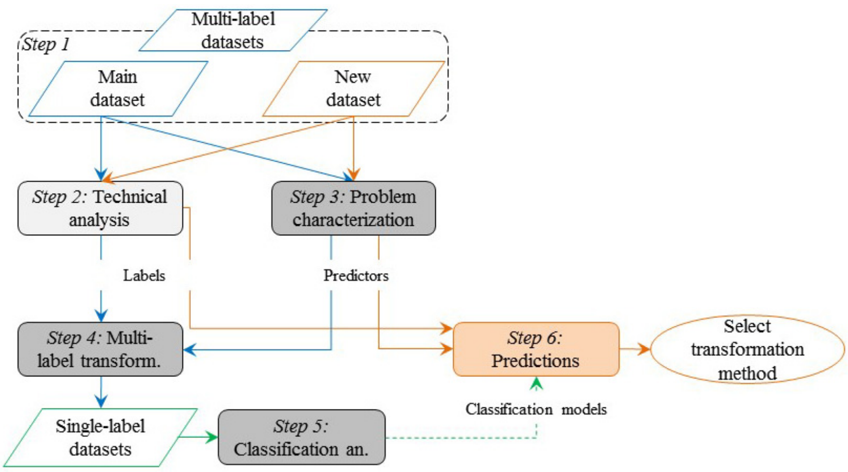

Fig. 1. Methodological framework

The methodological framework was developed considering the processing of two multi-label datasets (Step 1), named as main dataset and new dataset. Both datasets are generated to represent intrinsic problem characteristics. The main dataset (in blue) is used to fit classification models for each multi-label transformation method, converting the main dataset into single-label datasets (in green). The test dataset (in orange) is used to compare the accuracy performance of classification models fitted with each single-label dataset converted using multi-label transformation methods. A good accuracy performance is obtained when substantial information about intrinsic problem characteristics are not lost. Dashed arrows represent the transfer of fitted classification models throughout the framework and solid arrows are related to the input data transfer (labels, predictors, and datasets) to fit or use the classification models.

In technical analysis (Step 2), the objective is to obtain labels for each instance, represented by improvement heuristics. The 2D-SPP is a multi-label problem. For one instance, if the value of the best solution found is the same for more than one improvement heuristic, then this instance can be represented by more than one label.

Therefore, a constructive heuristic must be adopted to position the rectangles into the strip. The bottom-left (BL) is one of the most widely used constructive heuristics in the literature and was the first constructive heuristic developed based on the rectangles positioning concept (Baker et al., 1980). The BL aims to position each rectangle as lowest-left location as possible in any feasible space available in the strip until all rectangles are positioned. Thus, a complete solution with a strip height (H_{it}) is obtained. The great advantage of BL is its low complexity and very fast computational processing time.

In a second moment, a total of six improvement heuristics (completely random, dynamic random, random weight, tabu search, simulated annealing, and genetic algorithm) are used to improve the quality of the complete solution. At each iteration, changes in the rectangles sequence ordering are promoted according to the improvement heuristic rules. The best solution found using an improvement heuristic is given by the lowest gap (gap_{it}) between the strip height (H_{it}) found in each iteration (it) and the simple lower bound (LB_{it}), proposed by (Martello et al., 2003), as shown in (1).

$$gap_{it} = \frac{H_{it} - LB_{it}}{LB_{it}}. \tag{1}$$

Due to the intrinsic characteristics, an instance can be better adapted to a specific improvement heuristic, reflecting in different strip heights and gaps results. Therefore, the label of each instance is defined by the improvement heuristic that reached the lowest gap. If the lowest gap is given by more than one improvement heuristic, then the instance can be considered as multi-label.

In problem characterization (Step 3), the independent variables are predictors based on variations of problem characteristics. A total of 19 predictors (Table 1) were developed by (Neuenfeldt et al., 2019) for the 2D-SPP, based on information collected from cutting and packing problems generators and with an observation of literature instances. Variations in the rectangles and the strip dimensions, in addition to intrinsic instances information, were used to develop the 19 predictors (Neuenfeldt et al., 2017).

Table 1
Predictors' definition. Source: (Neuenfeldt et al., 2019; Neuenfeldt et al., 2017)

Predictor	Definition
areacomp	The ratio between the strip area and rectangles area, influenced by the ratio (between percentiles or quartiles measures) and composition (between the sum of larger and smaller measures) variables.
areastats	The ratio between the strip area and rectangles area, based on variables characterized by classical statistical measures (mean, median and, standard deviation).
perimcomp	The ratio between the strip perimeter and rectangles perimeters, influenced by ratio and composition variables.
perimstats	The ratio between the strip perimeter and rectangles perimeters, based on variables characterized by classical statistical measures.
dimcomp	The average dimension of rectangles compared to the strip width, influenced by ratio and composition variables.
dimstats	The average dimension of rectangles compared to the strip width, based on variables characterized by classical statistical measures.
widthdimcomp	Size of the largest rectangles dimension compared to the strip width, influenced by ratio and composition variables.
widthdimstats	Size of the largest rectangles dimension compared to the strip width, based on variables characterized by classical statistical measures.
propcomp	Level of proportion between the strip and rectangle's dimensions, influenced by ratio and composition variables.

Table 1 [cont.]

Predictor	Definition
propstats	Level of proportion between the strip and rectangles dimensions, based on variables characterized by classical statistical measures.
n	The total number of rectangles.
coefficient	Average rectangles' dimensions values.
heterog	The proportion of different rectangles.
heterognt	The proportion of different rectangles with more than one rectangle.
difcoefficient	The total number of different rectangles dimensions.
objdimratio	The number of times that the strip lower bound is bigger than the strip width.
itdimratio	The number of times that the maximum rectangles dimension is bigger than the minimum rectangles dimensions.
maxcoefficient	10% larger rectangles dimensions values.
mincoefficient	10% smaller rectangles dimensions values.

After identifying the predictors and labels for each instance, the multi-label dataset can be transformed into single-label datasets (Step 4), being able to fit classification models using SML techniques. Single-label datasets are generated for each transformation method.

The classification analysis (Step 5) should be developed by submitting the single-label datasets to different SML techniques, to verify which transformation method is more accurate with an acceptable level of generalization for the 2D-SPP. The use of more than one SML technique allows to check the behavior of a single-label dataset holistically and without bias. Thus, a pre-test with some instances was conducted to choose a total of six SML techniques (random forest, support vector machine, back-propagation neural networks, stochastic gradient boosting, extreme gradient boosting, and sparse partial least squares).

To fit classification models using SML techniques, a 5-fold cross-validation was developed, subdividing each single-label dataset into five folders. The best classification model of each SML technique is given by the highest accuracy classification model tested in all folders of each single-label dataset.

In predictions (Step 6), the fitted classification models for each single-label dataset are used to predict the label of the new dataset instances. Thus, the best transformation methods are given by the level of accuracy, calculated by the proportion of the number of instances in which the prediction corresponds to the label of each instance calculated by the improvement heuristic that reached the lowest gap. The more accurate the classification model is, the closer to 1 is the accuracy level.

Multi-label transformation methods

The literature review was established to show multi-label transformation methods. At the end of this section, some studies on transformation methods applied in the context of other problems are described.

As a meta-learning process, the classification analysis for the 2D-SPP can be solved using SML techniques. In stages, the machine adjusts a meta-model (named as "model") to select algorithms (improvement heuristics) based on problem characteristics provided by the instances dataset. As output, new instances datasets are submitted for the adjusted classification models to select heuristics for each new instance tested.

Multi-label transformation methods are used to convert multi-label instances into one or more single-label instances, allowing the use of traditional SML techniques. Over the years, different multi-label transformation methods have been developed for a wide range of problems. Below, six of the main multi-label transformation methods are presented.

In the first transformation method, named method 1 (ML1), a decomposition of multi-labels into several single-labels is proposed. No instances are lost and no original data is changed (Tsoumakas & Katakis, 2007). Fig. 2 shows an example where the decomposition promotes a significant expansion of the original dataset, which can affect the processing time required to fit classification models. Also, the decomposition allows all labels to be considered, even for multi-label instances. However, the characteristics of these instances are unique, which can cause a duality to fit more accurate classification models.

Instance	Label
1	A
2	A,B,C
3	C
4	A,C
5	B

Instance	Label
1	A
2	A
2	B
2	C
3	C
4	A
4	C
5	B

Fig. 2. Example of ML1 transformation method

In the exclusion method, named Method 2 (ML2), instances with multi-label behavior are not considered. All instances with more than one label are eliminated, avoiding stipulating a positive or negative interpretation bias for a specific label. Such bias could contribute to distortions in the pattern of the characteristics of the multi-label instance, affecting the classification models fitted (Tsoumakas & Katakis, 2007; Kanda et al., 2011). As shown in Fig. 3, the size of the generated dataset can decrease significantly compared to the original multi-label dataset.

Instance	Label
1	A
2	A,B,C
3	C
4	A,C
5	B

Instance	Label
1	A
3	C
5	B

Fig. 3. Example of ML2 transformation method

A negative aspect of the exclusion method is the loss of intrinsic problem characteristics to the excluded multi-label instances. The fitted classification models will not be able to recognize specific patterns verified in these instances. Also, if the original dataset has a high number of multi-label instances, then the single-label dataset will be composed of a very small number of instances, which may not accurately represent characteristics of the problem and, as a consequence, the classification models fitted are unable to accurately recognize cause-effect patterns (Kanda et al., 2011; Aleksovski et al., 2009).

The powerset method, named Method 3 (ML3), considers each multi-label instance as a new single-label option, as can be seen in Fig. 4. As a positive aspect, the results for each label are processed in detail. However, the possible low number of instances characterizing a new label can hinder the search for "usable" and representative problem patterns to fit classification models (Boutell et al, 2004). Another issue is related to the possibility of a future new instance being represented by a new label. Further analysis to verify the real label that best characterizes the instance is necessary.

Instance	Label
1	A
2	A,B,C
3	C
4	A,C
5	A,B,C

Instance	Label
1	A
2	Z1
3	C
4	Z2
5	Z1

Fig. 4. Example of ML3 transformation method

In the choice method, named Method 4 (ML4), for each multi-label instance, a unique label must be determined randomly, while the other labels are discarded, as shown in Fig. 5. In ML4 any instance is eliminated, preventing loss some problem characteristics. However, the random choice increases the difficulty to find patterns to fit classification models (Tsoumakas & Katakis, 2007).

Instance	Label
1	A
2	A,B,C
3	C
4	A,C
5	B

Instance	Label
1	A
2	C
3	C
4	A
5	B

Fig. 5. Example of ML4 transformation method

Finally, in the pseudo-label method, named Method 5 (ML5), the artificial creation of a new label for multi-label instances is necessary (Fig. 6). However, the pseudo-label may not accurately represent the instances' characteristics, confusing the verification of patterns and cause-effect relations between predictors variation and the label (or pseudo-label) to fit classification models. As in ML3, further analysis to verify the true label that best characterizes the instance based on the problem characteristics is necessary.

Instance	Label
1	A
2	A,B,C
3	C
4	A,C
5	A,B

Instance	Label
1	A
2	Z
3	C
4	Z
5	Z

Fig. 6. Example of ML5 transformation method

As mentioned before, problems with the multi-label format are found in different scientific research areas. However, no occurrence in the literature for the 2D-SPP or even for cutting and packing problems has been identified. A small review of articles from different research areas (as combinatorial optimization problem) were used to show the usage of multi-label transformation methods and to develop our methodological framework.

Reference (Kanda et al., 2011) used multi-label classification methods for the traveling salesman problem (TSP), in which the cost to travel between all cities and return to the starting point must be minimized. A model to predict the metaheuristics performance for new TSP instances is proposed, selecting the most promising optimization technique for a new TSP instance. Three transformation methods

(ML1, ML2, and the binary method) were used. For two datasets, synthetic and real data, the best performance was obtained using the binary method. Also, the results demonstrate that the problem for the TSP context is promising.

Reference (Dantas & Pozo, 2018) developed a comparison between a problem transformation method and an algorithm adaptation method to know the most efficient prediction for the quadratic assignment problem. Given two sets of locations and installations, the objective is to assign each installation to a unique location to minimize the total flow and the distance between associations. The ML3 was used with a total of 135 instances. As a result, both ML3 and algorithm adaptation methods achieved good performance, but the algorithm adaptation method was better to select algorithms for the quadratic assignment problem. Reference (Glinka et al., 2016) implemented multi-label transformation methods to facilitate the medical children diagnostics, using a multi-perspective classification problem as reference. A total of four multi-label transformation methods applied to real datasets were used, such as binary method and ML3. To find results, six datasets were separated, covering 2126 cases evaluated using two metrics: Accuracy and hamming loss.

Methodological framework usage

This section aims to describe the use of the methodological framework developed to select the multi-label transformation method. Firstly, the test input parameters and the generated instances are presented. Next, the improvement heuristics and the SML techniques are defined. Finally, the classification models obtained are tested and the accuracy is measured and analyzed for each transformation method.

The multi-label dataset is composed of 1000 instances, with a number of rectangles ranging between 7 and 2883, generated using 2DPackGen (Silva et al., 2014), based on input parameters from the 2D-SPP characteristics. The multi-label dataset was randomly divided into two parts. The training dataset is composed of 800 instances, 80% of the total, while the test dataset is composed of 200 instances, 20% of the total.

For technical analysis, six improvement heuristics were defined to calculate the lowest gap and assign a label to each instance. The improvement heuristics were performed using different criteria and parametrized specifications to order the rectangles sequence to be positioned into the strip, according to the improvement heuristics presented below. The criteria and parametrized specifications were established through extensive empirical studies conducted previously in this article.

The completely random, named Label A (LA), is a naive local search algorithm in which the rectangles sequence ordering is independent for each iteration and is defined randomly, exploring the solution space in a more general manner, being null the learning level by iteration. The dynamic random, named Label C (LC), randomly change the rectangles sequence ordering at each iteration by 5%. The dynamic random local search explores the solution space applying local and small changes, maintaining significant information about the current best solution.

For the random weight, named Label D (LD), in the first 4% of rectangles sequences are ordered by geometric characteristics as decreasing area, perimeter, width, and height. The next 76% of sequences are ordered using a random weighted procedure, divided into four parts containing changed sequences based on area, perimeter, width, and height. Finally, the last 20% of sequences are fully randomly generated. The rectangles sequence ordering is not completely random. Each rectangle has a probability to occupy the first place to be positioned based on the rectangles' geometric characteristics (Neuenfeldt et al., 2019). For LA, LC, and LD, the current reference solution is replaced only if the new solution found in each iteration is better (lowest gap).

The simulated annealing, named Label B (LB), changes the rectangles sequence ordering based on analogy to the thermodynamics cooling process. A solution can be accepted as a reference even if it is worse than the current reference solution, avoiding local optimal solutions. The initial temperature (T_o) in (2) is given by the rectangle's (r) area (A_r). The cooling temperature (T_{it}) at each iteration (it) is based on the initial temperature and two logarithmic functions, as shown in (3).

$$T_o = 0.05 \sum_{r=1}^{m} A_r \,, \tag{2}$$

$$T_{it} = \frac{T_0}{\left(\dfrac{\log k_{it}}{\log 2.717}\right)} \,; \quad it \leq 3 \rightarrow T_{it} = T_0 \,. \tag{3}$$

The temperature reduction factor (k_{it}) in (4) is a support parameter to measure the cooling intensity between iterations, based on the maximum number of iterations (M).

$$k_{it} = k_{it} + 2\left(\frac{100}{M}\right). \tag{4}$$

Finally, the acceptance probability (p_{it}) is the condition to assume a complete solution (H_{it}) as a new reference solution (H_b). If $p_{it} = 1$, H_{it} has a lower value than H_b and will be the new reference solution. Otherwise, p_{it} can be a value between 0 and 1, according to the acceptance probability function shown in (5).

$$p_{it} = \left[50\left(\frac{1}{H_{it} - H_b}\right) + T_{it}\right]\beta. \qquad (5)$$

If p_{it} is closer to zero, then the solution H_{it} is too far from H_b, which reduces the chances of H_{it} being the new reference solution. The auxiliary variable (β) is used to regulate how easily a worse quality solution will replace the current reference solution. A random number is generated at each iteration, if p_{it} is greater than this value, then H_{it} must replace H_b and be the new reference solution.

The genetic algorithm, named Label E (LE), is based on evolutionary biology, composed of a set of iterations in which new rectangles sequence ordering is obtained from parents' characteristics. As a genetic chromosome, in the crossover process, new sequences (children) receive parts of the parents' sequence ordering (Hartmann, 1998). The initial 30% of the new sequence is provided by parent 1 and the final 30% of the new sequence by parent 2. Also, the remaining intermediate new sequence (40%) is determined randomly to escape from the local optima, with sequences not selected from parents.

The 25 initial parents' generation is obtained using the rules and parameters defined for the dynamic random heuristic. At each iteration, five children are generated, with 80% of the children coming from the crossover and 20% generated using a completely random mutation method. The list of parents is updated through a probabilistic tournament selection, where rectangles sequence ordering with higher quality solutions are privileged. A total of five parents are replaced at each iteration.

In the tabu search, named Label F (LF), the information from previous iterations are memorized through a set of lists. At each iteration, prohibitions are introduced to not allow moving some rectangles in the sequence order. Two lists of prohibitions are used: the short-term and the long-term. For example, for instances with more than 200 rectangles, when selected for the short-term list, the rectangle cannot be moved in the next 8 consecutive iterations. If selected for the long-term list, the rectangle cannot be moved in the next 15 consecutive iterations. The values of the short-term and long-term lists can be increased or reduced proportionally to the total number of rectangles. For LE and LF, a complete solution will be a reference solution if the calculated height is lower than the current reference solution.

All improvement heuristics are run in the same environment, Windows 8.1 using an i7 3.0 GHz processor with 8 GB of RAM. This led to run times of approximately 900 seconds per instance. More details and concepts about improvement heuristics in the context of the 2D-SPP can be seen in (Oliveira et al., 2016).

The multi-label dataset is composed of 1000 instances. After solving all instances using the BL constructive heuristic and all 6 improvement heuristics to define the labels, a total of 871 instances are characterized as single-label, with the following distribution: LA (4); LB (23); LC (131); LD (84); LE (67); and LF (562). The remaining 129 instances are characterized as multi-label, where more than one quality solution was obtained. Fig. 7 shows the Venn diagram for all labels using the iterative tool for analysis proposed by (Heberle et al., 2015).

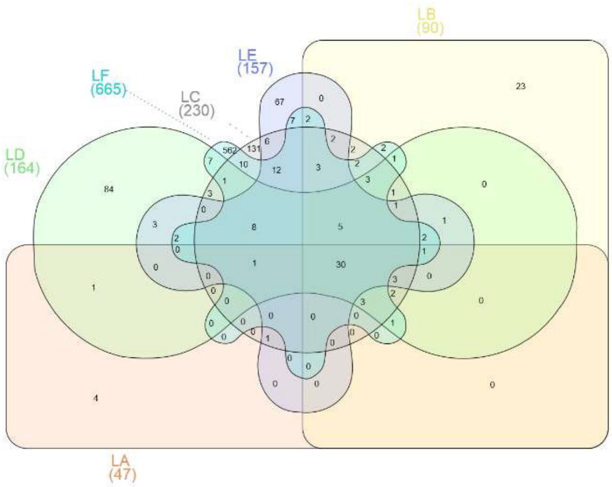

Fig. 7. Venn diagram for the multi-label dataset.

To fit classification models, six SML techniques were selected, preventing the analysis of the results under the bias of any SML technique specificity. The SML techniques used in this study are the random forest, support vector machine, back-propagation neural networks, stochastic gradient boosting, extreme gradient boosting, and sparse partial least squares.

The random forest combines the classification analysis developed by different decision trees to obtain a prediction with greater accuracy and stability, with the possibility to measure the importance of each predictor, where the most relevant are in nodes found in the first levels of the trees (Breiman, 2001). The extreme gradient boosting used a more regularized model formalization, which helps to reduce overfitting, providing a more efficient implementation of gradient boosting framework, observing the distribution of predictors in the decision trees to reduce search spaces (Chen, 1990; Chen & Tong, 2019). A gradient variable controls the size of the tree branches. The challenge is to find the best gradient value to avoid biases or generic classification models.

The support vector machine is one of the most flexible and effective techniques for classification analysis. The challenge is to efficiently adjust parameters of cost, order, and kernel, according to the problem characteristics (Suykens & Vandewalle, 1999). Backpropagation neural networks are composed of layers of input, output, and, at least, one layer with a nonlinear processing element (Chen, 1990). The technique works in two steps to train classification models: Feedforward and backward. In the feedforward step, the input data is processed by neurons. The output values are compared with known prediction values to estimate the results error. In the backward step, the error found is used to fit classification models. The algorithm continues to work in cycles to improve its learning capacity until a stop criterion is reached.

The stochastic gradient boosting has a constant $f(0 < f \leq 1)$ defined by the training dataset subsample size (Friedman, 2002). Smaller values of f introduce more randomness to avoid biased classification models, in addition to lower computational processing time. Instead, higher values of f can excessively adjust the classification model to the characteristics of the training dataset. Finally, the sparse partial least squares select predictors to fit classification models, using fit parameters and the sparsity concept (Chun & Keleş, 2010). Low sparsity values enable the selection of more predictors.

Table 2 shows the parameters used in each SML technique. The software RStudio, specifically the function "train" in "caret" package (Kuhn, 2008), was used to develop the classification analysis. The parameter values were defined based on the best classification models fitted to each single-label dataset and SML technique.

For the multi-label transformation process, five of the six transformation methods shown in Section 2 were used in this research (ML1, ML2, ML3, ML4, and ML5). Table 3 shows the results containing the accuracy level of the classification models fitted to

Table 2
Parameters used to fit classification models for each single-label dataset and SML technique

<table>
<tr><th>SML technique</th><th>Param.</th><th>ML1</th><th>ML4</th><th>ML2</th><th>ML3</th><th>ML5</th></tr>
<tr><td>Random forest (rf)</td><td>mtry</td><td colspan="5">2</td></tr>
<tr><td rowspan="3">Support vector machine (svmPoly)</td><td>degree</td><td colspan="3">1</td><td>2</td><td>1</td></tr>
<tr><td>scale</td><td colspan="3">0.001</td><td colspan="2">0.01</td></tr>
<tr><td>C</td><td colspan="2">0.25</td><td></td><td>1</td><td>0.5</td></tr>
<tr><td rowspan="2">Back-prop. neural networks (nnet)</td><td>decay</td><td>0</td><td colspan="2">0.0001</td><td colspan="2">0.1</td></tr>
<tr><td>size</td><td colspan="4">1</td><td>5</td></tr>
<tr><td rowspan="4">Stochastic gradient boosting (gbm)</td><td>n.trees</td><td colspan="5">50</td></tr>
<tr><td>iteration.depth</td><td colspan="5">1</td></tr>
<tr><td>shrinkage</td><td colspan="5">0.1</td></tr>
<tr><td>n.minobsinnode</td><td colspan="5">10</td></tr>
<tr><td rowspan="7">Extreme gradient boosting (xgbTree)</td><td>nrounds</td><td colspan="5">50</td></tr>
<tr><td>max_ depth</td><td colspan="5">1</td></tr>
<tr><td>eta</td><td colspan="5">0.3</td></tr>
<tr><td>gamma</td><td colspan="5">0</td></tr>
<tr><td>colsample bytree</td><td colspan="2">0.6</td><td colspan="3">0.8</td></tr>
<tr><td>min_ child_ weight</td><td colspan="5">1</td></tr>
<tr><td>subsample</td><td>1</td><td colspan="2">0.75</td><td>0.5</td><td>1</td></tr>
<tr><td rowspan="3">Sparse partial least squares (spls)</td><td>K</td><td>2</td><td>3</td><td>1</td><td colspan="2">7</td></tr>
<tr><td>eta</td><td>0.9</td><td>0.1</td><td colspan="2">0.5</td><td>0.1</td></tr>
<tr><td>kappa</td><td colspan="5">0.5</td></tr>
</table>

single-label datasets, as well as the total number of instances and labels. A standard accuracy behavior between SML techniques of each single-label dataset was obtained, demonstrating the reliability of the results, despite SML techniques having different characteristics to fit classification models using different ways.

Table 3
Accuracy results

SML technique	Accuracy				
	ML1	ML2	ML3	ML4	ML5
Random forest	0.40	**0.64**	0.57	0.57	0.58
Support vector machine	0.49	**0.65**	0.55	0.59	0.62
Back-propagation neural networks	0.48	**0.64**	0.56	0.59	0.60
Stochastic gradient boosting	0.46	**0.63**	0.56	0.58	0.60
Extreme gradient boosting	0.46	**0.63**	0.54	0.59	0.60
Sparse partial least squares	0.46	**0.64**	0.57	0.59	0.61
Mean	0.46	**0.64**	0.56	0.59	0.60
Number of instances	1353	**895**	1000	1000	1000
Number of labels	6	**6**	37	6	7

The strategy of decomposing the multi-labels in single-labels proposed in ML1 is not appropriated for the 2D-SPP, despite the strategy trying to keep most of the problem characteristics, without removing any instance from the dataset. A duality to represent the predictor's values of one instance for more than one single-label does not allow to fit classification models with good accuracy. As an example, the instance *pt1_99_325* has quality solutions for all labels, being divided into six single-label instances, each one with a different label, from LA until LF. To fit the classification model, any SML technique must read the same predictor's values of the instance *pt1_99_325* six times, with different labels.

In ML3, a total of 31 new labels were generated based on all multi-label instances combinations. The problem is the low representativeness of each new label, being infeasible to fit accurate classification models able to identify the problem characteristics of a new label. For example, both new labels, LBC and LBCD, are characterized by only one instance. Also, the excessive number of labels (6 proposed for the 2D-SPP +31 new) diffuse the input information about the problem provided to the SML techniques, fitting generic and unrepresentative classification models. Similarly, in ML5 the artificial creation of a pseudo-label for multi-label instances found is determined. For the seven labels (6 proposed for the 2D-SPP +1 pseudo), the ability to verify patterns to fit classification models was increased, mainly due to the improved pseudo-label instances representativeness compared to ML3. However, a new instance predicted with a pseudo-label cannot provide enough information about the best improvement heuristic option used to obtain the quality solution. Thus, the new instances predicted using a classification model provided by ML5 can be represented by a non-existent improvement heuristic.

For ML4, each multi-label instance is defined randomly by only one label. The other instance labels are excluded. The random selection process hindered the fitted classification models, due to the lack of efficiency to recognize patterns between problem characteristics given by the predictor's values and the label chosen for each multi-label instance.

In ML2, the ratio between the number of instances and the available labels is the lowest in comparison with other transformation methods. This fact can contribute to explaining the best accuracy obtained regardless of the SML technique used to fit classification models. The single-label dataset characteristics are less diffuse by the exclusion of all multi-label instances, allowing a real identification of the patterns between predictors and labels. However, a negative ML2 aspect can be a significant loss of information about the problem with the exclusion of instances from the main dataset, causing inaccuracy to predict labels of new instances with similar characteristics to those instances excluded. Also, this exclusion reduces the generalization level of the classification model, even the accuracy obtained with the ML2, in comparison to other transformation methods, is higher for the 2D-SPP.

Thus, a complementary study to verify the accuracy of the predictions only for the 129 multi-label instances excluded from the ML2 single-label dataset was conducted. The classification models fitted by the random forest and the support vector machine were used. If the predicted label matches with any multi-label of each instance, then the classification models are potentially accurate for multi-label instances.

For the random forest classification model, in 75% of instances, the predicted label is equal to one multi-label. For the support vector machine classification model, the accuracy is 80%. As expected, a reduced accuracy of 60% was verified for instances with two or three multi-labels, motivated by the reduction of the labels' assertiveness options. For more than four multi-labels, the accuracy is substantially improved. Therefore, for the 2D-SPP, the exclusion of multi-label instances did not significantly affect the ML2 ability to obtain single-label datasets and fit accurate and generalized classification models to different instances' characteristics.

Conclusions

As observed in the analysis of the results, the methodological framework works well to identify the transformation method for the 2D-SPP, providing coherent single-label datasets that can be used to fit classification models in future studies.

As mentioned in the introduction and shown during this article, the methodological framework proposed can be easily adapted. One of the basic premises and one of the main advantages of this research is the possibility to generalize the methodological framework for other combinatorial optimization problems variations (Neuenfeldt et al., 2022), mainly cutting and packing problems, as bin packing and knapsack problems. Few changes in the structure of the methodological framework are necessary to enable the use in other cutting and packing problems. As an example, for the bin packing problem, it is necessary to generate instances to be used as main datasets and new datasets, predictors to characterize the problem, and select constructive and improvement heuristics capable to solve the problem, thus obtaining labels for each main instance.

From an academic perspective, a lack of studies related to the context of multi-label transformations methods for cutting and packing problems and, in specific, for the strip packing problem was filled with this study, assisting other researchers in the use of data mining and machine learning approaches in the search for efficient solutions.

Some limitations were observed during the research. A limited number of constructive and improvement heuristics were used to calculate the gap and generate labels for each instance. Also, the problem was assigned to be solved by transformation methods, avoiding the use of adaptation methods. For future research, we expect to apply the methodological framework to transform multi-label instances from other cutting and packing problems in single-label instances. Also, the single-label dataset generated using the transformation method ML2 will be used to fit more detailed classification models and select improvement heuristics according to the characteristics of new instances for the 2D-SPP.

Acknowledgments

This work was supported by the Brazilian National Council for Scientific and Technological Development (CNPq) [grant number 422095/2018-4, 88887.522103/2020-00].

References

Aleksovski D., Kocev D. and Dzeroski S. (2009). *Evaluation of distance measures for hierarchical multilabel classification in functional genomics*. In 1st Workshop on Learning from Multi-Label Data (MLD) held in conjunction with ECML/PKDD, pp. 5–16.

Alvarez-Valdés R., Parreño F. and Tamarit J.M. (2008). Reactive GRASP for the strip–packing problem, *Computers & Operations Research*, vol. 35, no. 4, pp. 1065–1083. DOI: 10.1016/j.cor.2006.07.004.

Baker B.S., Coffman E.G. and Rivest R.L. (1980). Orthogonal packings in two dimensions, *SIAM Joruna*l *on Computing*, vol. 9, no. 4, pp. 846–855. DOI: 10.1137/0209064.

Boutell M.R., Luo J., Shen X. and Brown C.M. (2004). Learning multi-label scene classification, *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771. DOI: 10.1016/j.patcog.2004.03.009.

Brazdil P., Carrier C.G., Soares C. and Vilalta R. (2008). *Metalearning: Applications to data mining*. Springer Science & Business Media, Germany.

Breiman L. (2001). Random forests, *Machine Learning*, vol. 45, no. 1, pp. 5–32. DOI: 10.1023/A:1010933404324.

Chen F.-C. (1990). Back-propagation neural networks for nonlinear self-tuning adaptive control, *IEEE Control Systems Magazine*, vol. 10, no. 3, pp. 44–48. DOI: 10.1109/37.55123.

Chen, Tianqi, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen (2019). *xgboost: Extreme Gradient Boosting*. https://CRAN.R-project.org/package=xgboost.

Chun H. and Keleş S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection, *Journal of the Royal Statistical Society*, vol. 72, no. 1, pp. 3–25. DOI: 10.1111%2Fj.1467-9868.2009.00723.x.

Dantas A. and Pozo A. (2018). *Selecting Algorithms for the Quadratic Assignment Problem with a Multi-label Meta-Learning Approach*. In 2018 7th Brazilian Conference on Intelligent Systems (BRACIS). pp. 175–180. DOI: 10.1109/BRACIS.2018.00038.

Friedman J.H. (2002). Stochastic gradient boosting, *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378. DOI: 10.1016/S0167-9473(01)00065-2.

Glinka K., Wosiak A. and Zakrzewska D. (2016). *Improving children diagnostics by efficient multi-label classification method*. In Conference of Information Technologies in Biomedicine, pp. 253–266. DOI: 10.1007/978-3-319-39796-2_21.

Glover F. (1986). Future paths for integer programming and links to artificial intelligence, *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549. DOI: 10.1016/0305-0548(86)90048-1.

Hartmann S. (1998). A competitive genetic algorithm for resource-constrained project scheduling, *Naval Research Logistics*, vol. 45, no. 7, pp. 733–750. DOI: 10.1002/(SICI)1520-6750(199810)45:7<733::AID-NAV5>3.0.CO;2-C.

Heberle H., Meirelles G.V., da Silva F.R., Telles, G.P. and Minghim R. (2015). InteractiVenn: a web-based tool for the analysis of sets through Venn diagrams, *BMC bioinformatics*, vol. 16, no. 1, p. 169. DOI: 10.1186/s12859-015-0611-3.

Horvath M. and Vircikova E. (2012). Data mining model for quality control of primary aluminum production process, *Management and Production Engineering Review*, vol. 3, no. 1, pp. 47–53. DOI: 10.2478/v10270-012-0033-x.

Hopper E. and Turton B.C.H. (2001). An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem, *European Journal of Operational Research*, vol. 128, no. 1, pp. 34–57. DOI: 10.1016/S0377-2217(99)00357-4.

Kanda J., Carvalho A., Hruschka E. and Soares C. (2011). Selection of algorithms to solve traveling salesman problems using meta-learning, *International Journal of Hybrid Intelligent Systems*, vol. 8, no. 3, pp. 117–128. DOI: 10.3233/HIS-2011-0133.

Kuhn M. (2008). Caret package, *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26. DOI: 10.18637/jss.v028.i05.

Martello S., Monaci M. and Vigo D. (2003). An exact approach to the strip-packing problem, *INFORMS Journal on Computing*, vol. 15, no. 3, pp. 310–319. DOI: 10.1287/ijoc.15.3.310.16082.

Ntene N. and Vuuren J.H. (2009). A survey and comparison of guillotine heuristics for the 2D oriented offline strip packing problem, *Discrete Optimization*, vol. 6, no. 2, pp. 174–188. DOI: 10.1016/j.disopt.2008.11.002.

Neuenfeldt A.J., Silva E., Gomes A.M. and Oliveira J.F. (2017). *The two-dimensional strip packing problem: What matters?* In Congress of APDIO, the Portuguese Operational Research Society, pp. 151–164. DOI: 10.1007/978-3-319-71583-4_11.

Neuenfeldt A.J., Silva E., Gomes A.M., Soares C. and Oliveira J.F (2019). Data mining based framework to assess solution quality for the rectangular 2D strip-packing problem, *Expert Systems with Applications*, vol. 118, no. 1, pp. 365–380. DOI: 10.1016/j.eswa.2018.10.006.

Neuenfeldt A., Silva E., Francescatto M., Rosa C.B. and Siluk J. (2022). The rectangular two-dimensional strip packing problem real-life practical constraints: A bibliometric overview, *Computers & Operations Research*, vol. 157, no. 105521. DOI: 10.1016/j.cor.2021.105521.

Oliveira J.F., Neuenfeldt A.J., Silva E. and Carravilla M.A (2016). A survey on heuristics for the two-dimensional rectangular strip packing problem, *Pesquisa Operacional*, vol. 36, no. 2, pp. 197–226. DOI: 10.1590/0101-7438.2016.036.02.0197.

Rogalewicz M. and Sika R. (2016). Methodologies of knowledge discovery from data and data mining methods in mechanical engineering, *Management and Production Engineering Review*, vol. 7, no. 4, pp. 97–108. DOI: 10.1515/mper-2016-0040.

Silva E., Oliveira J.F. and Wäscher G. (2014). 2DCPackGen: A problem generator for two-dimensional rectangular cutting and packing problems, *European Journal of Operational Research*, vol. 237, no. 3, pp. 846–856. DOI: 10.1016/j.ejor.2014.02.059.

Suykens J.A.K. and Vandewalle J. (1999). Least squares support vector machine classifiers, *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300. DOI: 10.1023/A:1018628609742.

Tsoumakas G. and Katakis I. (2007). Multi-label classification: An overview, *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13. DOI: 10.4018/jdwm.2007070101.

Wäscher G., Haußner H. and Schumann H. (2007). An improved typology of cutting and packing problems, *European Journal of Operational Research*, vol. 183, no. 3, pp. 1109–1130. DOI: 10.1016/j.ejor.2005.12.047.