UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Irregular and quasi-polyomino strip packing problems

Marcos Okamura Rodrigues

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Marcos Okamura Rodrigues

Irregular and quasi-polyomino strip packing problems

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Franklina Maria Bragion de Toledo

USP – São Carlos
May 2020

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R685i Rodrigues, Marcos Okamura
 Irregular and quasi-polyomino strip packing
 problems / Marcos Okamura Rodrigues; orientadora
 Franklina Maria Bragion de Toledo. -- São Carlos,
 2020.
 90 p.

 Tese (Doutorado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2020.

 1. Strip packing. 2. Nesting. 3. Irregular
 shapes. 4. Quasi-polyomino. 5. Polyomino. I.
 Toledo, Franklina Maria Bragion de, orient. II.
 Título.

Marcos Okamura Rodrigues

Problemas de empacotamento em faixa de itens irregulares e quasi-poliominós

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Franklina Maria Bragion de Toledo

USP – São Carlos
Maio de 2020

To my parents, Antonio and Keiko.

ACKNOWLEDGEMENTS

To my master and doctoral advisor, Prof. Dr. Franklina Toledo, for the ideas, questions, critics and suggestion discussed in reunions that turn into this thesis.

To my TCC advisor, Prof. Dr. Adilson Bonifacio, for the collaboration, dedication and availability during the research.

To my scientific initiation advisor and my TCC coadvisor, Prof. Dr. Robinson Hoto, for the introduction to the research in optimization.

To my parents, Antonio and Keiko, the main people responsible for my education and formation.

To my brother Fernando and my sister Danielli to support me in my academic life.

To my friend and classmates at University of Sao Paulo (USP), especially to the member of the Laboratory of Optimization (LOt), to support me and motivate me during the graduation.

To my friend and classmates at Londrina State University (UEL), to support me and motivate me during the bachelor in Computer Science.

To every research who inspires me to follow the academic career.

To National Council for Scientific and Technological Development (CNPq), for the financial support in junior scientific initiation, scientific initiation and master scholarships.

To National Institute of Pure and Applied Mathematics (IMPA), for the development of the program of scientific initiation and Master (PICME).

To my ad-hoc adviser of FAPESP, for the collaboration and analysis and to send me his feedback as a volunteer.

To all examination professors, for attention and availability.

To everyone who help me somehow to the realization of this research.

To Sao Paulo Research Foundation (FAPESP), for the research grants #2014/23900-0 (doctoral) and #2013/14147-3 (master).

“If I have seen further it is by standing on the shoulders of Giants. .”
(Newton, Isaac)

RESUMO

RODRIGUES, M. O. **Problemas de empacotamento em faixa de itens irregulares e quasi-poliominós**. 2020. 90 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

O problema de empacotamento em faixa de itens irregulares consiste em cortar um conjunto de itens bidimensionais a partir de um objeto com largura fixa usando o menor comprimento possível. Apesar de sua importância econômica para várias indústrias, devido a sua dificuldade de resolução poucos métodos exatos foram direcionados para o problema. Recentemente, um modelo de programação inteira mista no qual os itens são posicionados sobre uma grelha foi proposto. Embora o modelo tenha provado a otimalidade para algumas instâncias de grande porte, ele possui um grande número de restrições de não-sobreposição, que cresce rapidamente de acordo com a resolução da discretização e o número de itens distintos. Nesta tese, é proposto um modelo de cobertura por cliques para reduzir o número de restrições e melhorar a relaxação linear. As coberturas são obtidas através de uma heurística desenvolvida pelo próprio autor. O modelo obtido superou a performance do modelo anterior para a maioria das instâncias avaliadas e obteve uma solução ótima para instância com até 25 itens (22 itens distintos) sujeito à discretização da grelha.

Recentemente, outro modelo de programação inteira mista foi proposto para o problema, mas ele permite um grande número de soluções simétricas. Nesta tese, novas restrições de quebra de simetria são propostas para melhorar o modelo. Experimentos computacionais foram realizados para instâncias com itens convexos. Os resultados indicaram que a formulação proposta é melhor que a anterior para a maioria das instâncias, uma vez que melhora os limitantes inferiores e reduz o tempo de execução e o número de nós explorados para provar a otimalidade.

Um caso particular de item irregular é um poliominó. Um poliominó consiste em um conjunto de quadrados de mesma dimensão conexos pela junção de uma de suas arestas. Um quasi-poliominó é uma generalização do conceito de poliominó, uma vez que representa um subconjunto de quadrados não necessariamente conexos de uma malha quadriculada equidistante. Problemas de corte e empacotamento de quasi-poliominós possuem diversas aplicações reais, por exemplo, o corte de itens de couro, a estamparia de chapas metálicas, o desenho de placas de circuito impresso e a diagramação de páginas de revistas e jornais. Nesta tese, estudamos o problema de empacotamento em faixa de quasi-poliominós. São propostos dois modelos de programação inteira para o problema e realizados testes computacionais para avaliá-los. Os modelos foram avaliados utilizando instâncias da literatura e apresentaram bons resultados, obtendo uma solução ótima para uma instância com 320 itens (20 itens distintos) em um recipiente de dimensões 44x50. Como esperado, foram encontradas mais soluções ótimas quando não há rotações e reflexões e quando as dimensões dos itens são pequenas.

Palavras-chave: Corte e empacotamento, Quasi-poliominós, Itens irregulares, Empacotamento em faixa, Programação inteira mista.

ABSTRACT

RODRIGUES, M. O. **Irregular and quasi-polyomino strip packing problems.** 2020. 90 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

The irregular strip packing problem consists in the cutting of a set of two-dimensional pieces from an object of fixed width using the minimum possible length. Despite its economic importance for many industries, because of its resolution difficulty few exact methods have addressed this problem. Recently, a mixed integer programming model in which pieces are placed on a grid has been proposed. Although the model has proved the optimality for some large instances, it has a large number of non-overlap constraints, which grows quickly according to the discretization resolution and number of distinct pieces. This thesis proposes a clique covering model to reduce the number of constraints and improve the linear relaxation. The coverings were obtained by a heuristic developed by the author. The model has outperformed the previous model in most evaluated instances and obtained an optimal solution for instances with up to 25 pieces (22 distinct pieces) subject to grid discretization.

Recently, another mixed integer programming model was proposed for the problem, but it may allow a large number of symmetric solutions. In this thesis, new symmetry breaking constraints are proposed to improve the model. Computational experiments were performed for instances with convex pieces. The results show the proposed formulation is better than the previous one for most instances, since it improves lower bounds and reduces run-time and number of nodes explored to prove optimality.

A particular case of an irregular item is a polyomino. A polyomino is a set of unit squares connected by joining one of their edges. A quasi-polyomino is a polyomino generalization, since it is a subset of not necessarily connected squares obtained from an equidistant raster grid. Quasi-polyomino cutting and packing problems have many real applications, e.g., leather cutting, sheet metal stamping, design of printed circuit boards and layout of magazines and newspapers. In this thesis, we study the quasi-polyomino strip packing problem. We propose two integer programming models for the problem and evaluate them using state-of-the-art solvers. We evaluate the models using instances taken from the literature and both models obtained good results, solving to optimality an instance with 320 items (20 distinct items) on a strip of dimensions 44x50. As expected, we found more optimal solutions when there are no rotation and flips and when the dimensions of the items are small.

Keywords: Cutting and packing, Quasi-polyominoes, Irregular shapes, Strip packing, nesting, Mixed integer programming.

LIST OF FIGURES

Figure 1 – An irregular strip packing problem.	2
Figure 2 – A polyomino, a pseudo-polyomino and a quasi-polyomino.	4
Figure 3 – Representation of a piece type.	12
Figure 4 – Example of inner-fit polygon in the grid.	12
Figure 5 – Construction of the nofit polygon in the grid.	13
Figure 6 – Representation of a piece type.	14
Figure 7 – Example of an inner-fit polygon on the grid.	15
Figure 8 – Construction of a nofit polygon on the grid.	15
Figure 9 – Example of an irregular strip packing problem.	17
Figure 10 – Nofit polygons for pieces 1 and 2.	17
Figure 11 – Example of a conflict graph.	17
Figure 12 – Representation of a piece and its distance parameters in NFP-CM.	30
Figure 13 – Non-overlap regions.	32
Figure 14 – A polyomino, a pseudo-polyomino and a quasi-polyomino.	37
Figure 15 – Placement of an item in the bin using Beasley (1985) model.	39
Figure 16 – Calculation of objective function in the model of Kashkoush, Shalaby and Abdelhafiez (2012).	42
Figure 17 – Example of solutions for the problem that minimize the penalties of used cells	42
Figure 18 – A quasi-polyomino strip packing problem.	42
Figure 19 – Representing rotation and flip of a quasi-polyomino.	43
Figure 20 – Overlap verification and objective function calculation for a cell of the strip.	45
Figure 21 – Number of binaries, variables and constraints for Kashkoush et al. and Strip Sum models (in logscale).	52

LIST OF TABLES

Table 1 – Features of data instances.	22
Table 2 – Trivial and iterative lower bounds.	24
Table 3 – Statistics of the clique covering and dotted-board models.	25
Table 4 – Comparison between clique covering and dotted-board models (time limit of 18,000 s).	26
Table 5 – Comparison between clique covering and dotted-board models (time limit of 3,600 s).	27
Table 6 – Features of benchmark instances.	35
Table 7 – Comparison between MIP models.	36
Table 8 – Number of constraints and variables of the proposed models.	47
Table 9 – Features of the evaluated instances.	50
Table 10 – Results obtained for the strip packing problem.	51
Table 11 – Results obtained for the strip packing problem - random instances without rotations and flips (part 1).	53
Table 12 – Results obtained for the strip packing problem - random instances with rotations and flips (part 1).	54
Table 13 – Results obtained for the strip packing problem - random instances without rotations and flips (part 2).	54
Table 14 – Results obtained for the strip packing problem - random instances with rotations and flips (part 2).	54

CONTENTS

1	INTRODUCTION	1
1.1	The irregular strip packing problem	1
1.2	Quasi-polyomino tiling and strip packing problems	3
1.3	Thesis contributions	5
1.4	Thesis organization	5
1.4.1	<i>A clique covering MIP model for the irregular strip packing problem</i>	6
1.4.2	<i>MIP models for the irregular strip packing problem: new symmetry breaking constraints</i>	6
1.4.3	<i>Integer programming models for the quasi-polyomino strip packing problem</i>	6
2	A CLIQUE COVERING MIP MODEL FOR THE IRREGULAR STRIP PACKING PROBLEM	9
2.1	Introduction	9
2.1.1	<i>Toledo et al. (2013) model</i>	11
2.2	Problem definition	13
2.3	Clique covering model	16
2.3.1	<i>Non-overlap constraints</i>	16
2.3.2	<i>Objective function</i>	18
2.3.3	<i>Proposed Model</i>	19
2.3.4	<i>Valid inequalities and lower bound</i>	19
2.4	Computational experiments	20
2.5	Conclusions and further work	28
3	MIP MODELS FOR THE IRREGULAR STRIP PACKING PROBLEM: NEW SYMMETRY BREAKING CONSTRAINTS	29
3.1	Introduction	29
3.2	Cherri et al. (2016) model	30
3.3	MIP model	30
3.4	Symmetry breaking constraints	31
3.4.1	<i>Partitioning the solution space</i>	31
3.4.2	<i>Partitioning the solution space</i>	32
3.4.3	<i>Connecting pieces</i>	32

3.4.4	<i>Connecting pieces with the board</i>	33
3.4.5	<i>Pieces with same shape</i>	34
3.5	Results	34
3.6	Conclusions and further work	35
4	INTEGER PROGRAMMING MODELS FOR THE QUASI-POLYOMINO STRIP PACKING PROBLEM	37
4.1	Introduction	37
4.2	Beasley's model	39
4.3	Kashkoush, Shalaby and Abdelhafiez's model	40
4.4	Proposed model	42
4.4.1	<i>Representing the items and the strip</i>	43
4.4.2	<i>Problem constraints</i>	43
4.4.2.1	<i>Demand constraints</i>	44
4.4.2.2	<i>Non-overlap constraints</i>	44
4.4.3	<i>Strip_Max model</i>	45
4.4.4	<i>Strip_Sum model</i>	46
4.4.5	<i>Analysis of the models</i>	47
4.4.6	<i>Valid inequalities and variable fixing</i>	47
4.4.7	<i>Strip_Sum model</i>	48
4.5	Computational experiments	48
4.5.1	<i>Instances</i>	48
4.5.1.1	<i>Instance set 1</i>	48
4.5.1.2	<i>Instance set 2</i>	49
4.5.2	<i>Computational Results</i>	49
4.5.2.1	<i>Instance set 1</i>	49
4.5.2.2	<i>Instance set 2</i>	53
4.6	Conclusions and further work	53
5	FINAL REMARKS AND FURTHER WORKS	55
BIBLIOGRAPHY		57
APPENDIX A	SOLUTIONS - A CLIQUE COVERING MIP MODEL FOR THE IRREGULAR STRIP PACKING PROBLEM	65
APPENDIX B	SOLUTIONS - MIP MODELS FOR THE IRREGULAR STRIP PACKING PROBLEM: NEW SYMMETRY BREAKING CONSTRAINTS	73

APPENDIX C	SOLUTIONS - INTEGER PROGRAMMING MODELS FOR THE QUASI-POLYOMINO STRIP PACKING PROBLEM	77
------------	--	----

INTRODUCTION

Irregular cutting and packing problems are relevant for many industries, e.g., shoes, garment, furniture, metal, glass and paper industries. The problems are relevant, since they reduce the production costs and minimize raw material waste. Such problems are NP-hard (FOWLER; PATERSON; TANIMOTO, 1981; NIELSEN; ODGAARD, 2003) and combine the difficulty of cutting and packing problems with the geometric complexity of placing irregular items. In this sense, we must develop a set of tools to handle the geometry of the items, which is not a trivial task and may be a barrier to new research in the area (BENNEL; OLIVEIRA, 2008). This problem is detailed in Section 1.1

A particular case of the irregular strip packing problem is the quasi-polyomino strip packing problem. Quasi-polyomino cutting and packing problems have many applications, e.g., sheet metal stamping, design of printed circuit boards, timber cutting and page layout of newspaper (KASHKOUSH; SHALABY; ABDELHAFIEZ, 2012). These problems are described in Section 1.2

1.1 The irregular strip packing problem

The irregular strip packing problem consists in the cutting of a set of two-dimensional items from a board of fixed width and infinite length. The objective is to minimize the length of the board used to cut all items (see Figure 1). According to Wäscher, Haußner and Schumann (2007), the problem can be classified as open, two-dimensional and irregular.

Several authors have developed heuristics to solve the problem (Dowsland and Dowsland (1995), Bennell and Oliveira (2009) present a review). We can divide the methods into constructive and improvement heuristics. Most studies that employ constructive heuristics are based on bottom-left (OLIVEIRA; GOMES; FERREIRA, 2000; DOWSLAND; VAID; DOWSLAND, 2002) and bottom-left-fill (BURKE *et al.*, 2006; BURKE *et al.*, 2010) algorithms. The algo-

Figure 1 – An irregular strip packing problem.

Source: Elaborated by the author.

gorithms sequentially place each item in the leftmost possible position on the board and in the bottommost position, in case of a tie. Hybrid algorithms (BENNEL; DOWSLAND, 2001; GOMES; OLIVEIRA, 2006), guided local search (EGEBLAD; NIELSEN; ODGAARD, 2007; UMETANI *et al.*, 2009), iterated local search (IMAMICHI; YAGIURA; NAGAMOCCHI, 2009), beam search (BENNEL; SONG, 2010), extended local search (LEUNG; LIN; ZHANG, 2012), simulated annealing with collision free region (SATO; MARTINS; TSUZUKI, 2012), cuckoo search (ELKERAN, 2013) and genetic algorithm (PINHEIRO; AMARO JR; SARAIVA, 2015; AMARO JR; PINHEIRO; COELHO, 2017; MUNDIM; ANDRETTA; QUEIROZ, 2017) are some of the algorithms applied to solve the problem.

The best solutions for benchmark instances for the problem were obtained by (ELKERAN, 2013). The authored developed a guided cuckoo search algorithm to solve the problem. In the method, the best fit between two items is found using a pairwise clustering and then the guided cuckoo search algorithm is applied to obtain complete solutions.

Recently, Leao *et al.* (2020) present a review of mathematical models for the irregular strip packing problem. The first exact study to the irregular strip packing problem was based on constraint programming. Ribeiro, Carravilla and Oliveira (1999) created a search strategy to solve the strip packing problem with only convex items. The initial variable domains were defined based on the allowable points where the reference point of each polygon can be placed. For each pair of items, they imposed a non-overlap constraint based on the nofit polygon. Carravilla, Ribeiro and Oliveira (2003) extended the previous research to deal with nonconvex irregular items and developed symmetry breaking constraints to deal with items of the same shape. In both studies, the variables associated with the items' reference point belonged to a discrete domain. The authors obtained an optimal solution for an instance with seven items, which is subject to discretization. Cherri *et al.* (2019) proposed a constraint programming model with NoOverlap global constraint for the problem. The computational experiments show that the CP model can quickly find good quality solutions in comparison to state-of-the-art models.

Fischetti and Luzzi (2009) created the first mixed integer programming model for the problem based on Daniels, Li and Milenkovic (1994) and Li (1994). It has continuous variables to define the position of each item and binary variables to prevent overlap. The authors imposed

the non-overlap constraints by placing the reference point of the item inside one of the partitions associated with the nofit polygon complement defined by each pair of items. The proposed model can solve instances with up to seven items to optimality.

Alvarez-Valdes, Martinez and Tamarit (2013) developed a procedure to obtain the partitions described by Fischetti and Luzzi (2009) via horizontal slices. They extended the linear compaction model proposed by Gomes and Oliveira (2006) to handle the strip packing problem with convex items and reformulated the Fischetti and Luzzi (2009) model, lifting both lower and upper bounds of the continuous variables. They also developed a branch-and-bound algorithm that proved optimality for instances with up to 16 items.

Toledo *et al.* (2013) created the dotted-board model, a mixed integer programming model where the board is represented by a grid. Binary variables are associated with a item type and a grid dot, where the items can be placed inside the board. The authors solved instances with up to 16 items (four types), 21 items (seven types), 35 items (one type) and 56 items (two types) to optimality and the solutions were subjected to grid discretization. Although the model can solve instances with many items, it has two main drawbacks: it uses a large number of binary variables to describe the grid and imposes a huge number of constraints to prevent overlap.

The most recent models to irregular strip packing problem are Cherri *et al.* (2016), Leao *et al.* (2016), Santoro and Lemos (2015). To combine the accuracy of continuous models with the advantages of the dotted-board model, Leao *et al.* (2016) proposed a semi-continuous model where the items can be placed in continuous positions in the x-axis and in discrete positions in the y-axis. Santoro and Lemos (2015) proposed a MIP model to convex polygons with some limitations: the polygons can have up to eight edges and their opposite edges must be parallel. Cherri *et al.* (2016) proposed two models that allow rotation of items. The models differ in the geometric structures used to avoid overlap among items.

1.2 Quasi-polyomino tiling and strip packing problems

Golomb (1954) defined the concepts of polyomino, pseudo-polyomino and quasi-polyomino based on a chessboard. A polyomino or n -omino is a set of n squares from the board connected by their edges, like the movement of a rook (rook-wise connected). Similarly, a pseudo-polyomino is a set of squares from the board connected by their edges or vertices, like the movement of a queen (queen-wise connected). A quasi-polyomino is simply a set of squares from the board. Figure 14 illustrates (a) a polyomino, (b) a pseudo-polyomino and (c) a quasi-polyomino. Note that every polyomino is a pseudo-polyomino and every pseudo-polyomino is a quasi-polyomino.

The quasi-polyomino strip packing problem consists in cutting a set of items (quasi-polyominoes) from a strip with fixed width and infinite length. The objective is to minimize the length required to cut all items. According to the typology proposed by Wäscher, Haußner and Schumann (2007), the problem is two-dimensional with one open dimension and irregular shapes.

Figure 2 – A polyomino, a pseudo-polyomino and a quasi-polyomino.

Source: Elaborated by the author.

The problem is NP-hard, since the two-dimensional strip packing problem ([LODI; MARTELLO; MONACI, 2002](#)) can be reduced to it, using rectangular items with integer dimensions.

Another cutting and packing problem is the tiling problem. The tiling problem is a feasibility problem that consists in checking whether a set of items can completely cover a board with fixed dimensions. According to the typology proposed by [Wäscher, Haußner and Schumann \(2007\)](#), the quasi-polyomino tiling problem is two-dimensional with irregular items. The problem is NP-complete, since the two-dimensional tiling problem with rectangles ([FOWLER; PATERSON; TANIMOTO, 1981](#)) can be reduced to the polyomino tiling problem.

Several heuristic methods have been proposed for the irregular strip packing problem via an approximation of irregular shapes by quasi-polyominoes. [Pargas and Jain \(1993\)](#) developed a parallel stochastic optimization algorithm, while [Babu and Babu \(2001\)](#), [Fischer and Dagli \(2004\)](#), [Wong et al. \(2009\)](#), [Zhang and Yang \(2009\)](#), [Siasos and Vosniakos \(2014\)](#) developed genetic algorithms for the irregular strip packing problem using approximations by polyominoes. In general, these algorithms have the same basic structure. Each individual of the population is represented by a sequence of items and eventual rotations and/or flips, which is used as input for a constructive heuristic, usually the bottom-left algorithm. [Shalaby and Kashkoush \(2013\)](#) proposed a particle swarm optimization algorithm for the irregular strip packing problem. The algorithm had two stages, the first used an approximation by polyominoes to place the items and the second performed a compaction heuristic based on direct trigonometry. [Baldacci et al. \(2014\)](#) employed a lagrangian heuristic based on an approximation by polyominoes to solve real problem instances from leather cutting.

Several authors also have developed heuristic algorithms for the polyomino tiling problem. [Takefuji and Lee \(1990\)](#), [Asai, Nakayama and Ninomiya \(1996\)](#), [Manabe and Asai \(1999\)](#) used optimization algorithms based on neural networks to solve the problem. [Gwee and Lim \(1996\)](#) developed a genetic algorithm for the problem where the items are placed on the corners of the board in a circular sequence. [Merlini, Sprugnoli and Verri \(2000, 2002\)](#) proposed algorithms that transform a tiling problem into an equivalent regular grammar, i. e., all the possible solutions are

represented by a regular language. [Amintoosi *et al.* \(2007\)](#) used pattern matching to deal with the problem based on the frontiers (edges) of the items.

On the other hand, few exact approaches have been proposed for quasi-polyomino packing problems. [Kashkoush, Shalaby and Abdelhafiez \(2012\)](#) developed a mixed integer programming model for both polyomino strip packing and tiling problems, considering one-sided polyomino (polyomino without flips) with rotations of 0 and 180 degrees.

1.3 Thesis contributions

The contribution of this thesis is the development of better mathematical models for the irregular and quasi-polyomino strip packing problems. The main contributions are described below.

1. Two models are proposed for the irregular strip packing problem: clique covering model and improved NFP-CM. Both models solved larger instances than state-of-the-art models proposed by [Toledo *et al.* \(2013\)](#) and [Cherri *et al.* \(2016\)](#). The clique covering model obtained an optimal solution for instances with up to 25 items (22 distinct items) subject to grid discretization. The results show the improved NFP-CM is better than NFP-CM for most instances, since it improves lower bounds and reduces run-time and number of nodes explored to prove optimality. This occurs because of the addition of the symmetry breaking constraints.
2. Two models are proposed for the quasi-polyomino strip packing problem: Strip_Sum and Strip_Max. Both models obtained better results than the model proposed by [Kashkoush, Shalaby and Abdelhafiez \(2012\)](#). Strip_Sum and Strip_Max models solved to optimality instances with up to 320 items (20 distinct items) on a strip of dimensions 44x50.

1.4 Thesis organization

In this section we present the organization of this thesis. The clique covering model for the irregular strip packing problem is presented in Chapter 2. The symmetric breaking constraints for the irregular strip packing problem are described in Chapter 3. The integer programming models for the quasi-polyomino strip packing problem are presented in Chapter 4. The final considerations and further works can be found in Chapter 5. The solutions obtained in this thesis can be found in Appendices A - C.

This thesis has generated one full paper and one short paper described in Chapters 2 and 3, respectively. The abstracts and references of the papers are presented below.

1.4.1 *A clique covering MIP model for the irregular strip packing problem*

This paper is available online at Computers & Operations Research website <<http://dx.doi.org/10.1016/j.cor.2016.11.006>>. RODRIGUES, M. O.; TOLEDO, F. M. B. *A clique covering MIP model for the irregular strip packing problem*. *Computers & Operations Research*. Volume 87, 2017, p. 221-234. [Rodrigues and Toledo \(2017\)](#).

Abstract: The irregular strip packing problem consists in the cutting of a set of two-dimensional pieces from an object of fixed width using the minimum possible length. Despite its economic importance for many industries, few exact studies have addressed this problem. Recently, a mixed integer programming model in which pieces are placed on a grid has been proposed. Although the model has proved the optimality for some large instances, it has a large number of non-overlap constraints, which grows quickly according to the discretization resolution and number of distinct pieces. This paper proposes a clique covering model to reduce the number of constraints and improve the linear relaxation. The model has outperformed the previous model in most evaluated instances and obtained an optimal solution for instances with up to 25 pieces (22 distinct pieces) subject to grid discretization.

1.4.2 *MIP models for the irregular strip packing problem: new symmetry breaking constraints*

This short paper is available online at APMOD 2016 conference proceeding website: <<https://doi.org/10.1051/itmconf/20171400005>>. RODRIGUES, M. O.; CHERRI, L. H.; MUNDIM, L. R. *MIP models for the irregular strip packing problem: new symmetry breaking constraints*. *ITM Web Conf.*, 14. p. 5. [Rodrigues, Cherri and Mundim \(2017\)](#).

Abstract: The irregular strip packing problem consists in minimizing the length used to cut a set of pieces from a board with fixed width. Recently, a mixed integer programming model was proposed for the problem, but it may allow a large number of symmetric solutions. In this paper, new symmetry breaking constraints are proposed to improve the model. Computational experiments were performed for instances with convex pieces. The results show the proposed formulation is better than the previous one for most instances, since it improves lower bounds and reduces run-time and number of nodes explored to prove optimality.

1.4.3 *Integer programming models for the quasi-polyomino strip packing problem*

This is a working paper.

Abstract: A polyomino is a set of unit squares connected by joining one of their edges. A quasi-polyomino is a polyomino generalization, since it is a subset of not necessarily connected

squares obtained from an equidistant raster grid. Quasi-polyomino cutting and packing problems have many real applications, e.g., leather cutting, sheet metal stamping, design of printed circuit boards and layout of magazines and newspapers. In this paper, we study the quasi-polyomino strip packing problem. The problem consists in cutting a set of quasi-polyominoes from a strip with fixed width and infinite length. The objective is to minimize the length required to cut all items. We propose two integer programming models for the problem and evaluate them using state-of-the-art solvers. We evaluate the models using instances taken from the literature and both models obtained good results, solving to optimality instances with up to 320 items (20 distinct items) on a strip of dimensions 44×50 . As expected, we found more optimal solutions when there are no rotation and flips and when the dimensions of the items are small.

A CLIQUE COVERING MIP MODEL FOR THE IRREGULAR STRIP PACKING PROBLEM

This chapter is based on the paper published and available online at Computers & Operations Research website <<http://dx.doi.org/10.1016/j.cor.2016.11.006>>. RODRIGUES, M. O.; TOLEDO, F. M. B. *A clique covering MIP model for the irregular strip packing problem*. *Computers & Operations Research*. Volume 87, 2017, p. 221-234. [Rodrigues and Toledo \(2017\)](#).

This research began during the author's master's degree and was concluded at his doctorate. The author developed an heuristic that is able to reduce the number of cliques to cover the edges of the overlap graph. In this sense, it can solve larger instances.

2.1 Introduction

Irregular cutting and packing problems arise in industries of shoes, garment, furniture, metal, glass and paper. Such problems are both economically and ecologically relevant, since they aim at reducing the production costs and minimizing raw material waste. They are also NP-hard ([FOWLER; PATERSON; TANIMOTO, 1981](#); [NIELSEN; ODGAARD, 2003](#)) and combine the difficulty of cutting and packing problems with the geometric complexity of placing irregular convex and nonconvex pieces. Consequently, a set of tools must be developed to deal with the geometry of the pieces, which is not a trivial task and may be a barrier to new research in the area ([BENNEL; OLIVEIRA, 2008](#)).

The irregular strip packing problem consists in the cutting of a set of two-dimensional pieces from a board of fixed width and infinite length. The objective is to minimize the length of the board used to cut all pieces. According to the typology proposed by [Wäscher, Haußner and Schumann \(2007\)](#), this problem can be classified as open, two-dimensional and irregular.

Several authors have developed heuristic methods to solve the problem (see [Dowsland](#)

and Dowsland (1995), Bennell and Oliveira (2009) for a review). In general, such methods can be divided into constructive and improvement heuristics. Most studies that employ constructive heuristics are based on bottom-left (OLIVEIRA; GOMES; FERREIRA, 2000; DOWSLAND; VAID; DOWSLAND, 2002) and bottom-left-fill (BURKE *et al.*, 2006; BURKE *et al.*, 2010) algorithms, which sequentially insert each piece in the leftmost possible position on the board and in the bottommost position, in case of a tie. Hybrid algorithms (BENNEL; DOWSLAND, 2001; GOMES; OLIVEIRA, 2006), guided local search (EGEBLAD; NIELSEN; ODGAARD, 2007; UMETANI *et al.*, 2009), iterated local search (IMAMICHI; YAGIURA; NAGAMOCHI, 2009), beam search (BENNEL; SONG, 2010), extended local search (LEUNG; LIN; ZHANG, 2012), simulated annealing with collision free region (SATO; MARTINS; TSUZUKI, 2012), cuckoo search (ELKERAN, 2013) and genetic algorithm (PINHEIRO; AMARO JR; SARAIVA, 2015) are some of the algorithms applied to tackle the problem.

The first exact approach to the irregular strip packing problem was based on constraint programming. Ribeiro, Carravilla and Oliveira (1999) developed a search strategy to solve the strip packing problem with convex pieces. The initial variable domains were defined in accordance with the allowable points where the reference point of each polygon can be placed. For each pair of pieces, they imposed a non-overlap constraint based on its respective nofit polygon. Carravilla, Ribeiro and Oliveira (2003) extended the previous study to deal with nonconvex irregular pieces and developed symmetry breaking constraints to handle pieces of same shape. In both approaches, the variables associated with the pieces' reference point belonged to a discrete domain. The authors found an optimal solution for an instance with seven pieces, which is subject to discretization.

Fischetti and Luzzi (2009) proposed the first mixed integer programming model for the problem based on Daniels, Li and Milenkovic (1994) and Li (1994). It uses continuous variables to define the position of each piece and binary variables to prevent overlap. The authors imposed the non-overlap constraints by positioning the piece's reference point inside one of the partitions associated with the nofit polygon complement defined by each pair of pieces. The model can solve instances with up to seven pieces to optimality.

Alvarez-Valdes, Martinez and Tamarit (2013) defined a procedure to obtain the partitions described by Fischetti and Luzzi (2009) using horizontal slices. They extended the linear compaction model proposed by Gomes and Oliveira (2006) to deal with the strip packing problem with convex pieces and reformulated the Fischetti and Luzzi (2009) model, lifting both lower and upper bounds of the continuous variables. They also developed a branch-and-bound method that proved optimality for instances with up to 16 pieces.

Toledo *et al.* (2013) proposed the dotted-board model, a mixed integer programming model in which the board is represented by a grid. Binary variables are associated with a piece type and a grid dot, where the pieces can be positioned inside the board. The authors solved instances with up to 16 pieces (four types), 21 pieces (seven types), 35 pieces (one type) and 56

pieces (two types) to optimality and the solutions were subjected to grid discretization. Although the model can solve instances with dozens of pieces, it has two main drawbacks: it uses a large number of binary variables to represent the grid and imposes a huge number of constraints to prevent overlap.

This paper proposes a clique covering model based on the dotted-board model. A huge number of constraints from the dotted-board model is replaced by a significantly smaller set. The auxiliary constraints used to define the solution length are replaced by a set based on a vertex clique covering and then, the non-overlap constraints are replaced by an edge clique covering. This approach reduces the number of constraints and nonzeros of the model and can improve its linear relaxation. A method to improve the lower bound of the model is also proposed. The results show the model has outperformed the dotted-board model and can solve instances with up to 22 distinct pieces to optimality, subject to the grid discretization.

The remainder of the paper is organized as follows: Section 2.2 defines the problem and reviews the dotted-board model proposed by Toledo *et al.* (2013); Section 4.4 proposes the clique covering model and introduces valid inequalities and lower bounds; Section 4.5 reports computational experiments for both models using instances taken from the literature; finally, Section 2.5 addresses the conclusions and suggests further works.

2.1.1 Toledo *et al.* (2013) model

Toledo *et al.* (2013) proposed a mixed integer programming model for the irregular strip packing problem where the strip is described by a grid. Given a strip with fixed width W and initial length L (upper bound), the objective of the problem is to minimize the length used z of the strip. In this sense, to obtain the grid, we make a horizontal and vertical discretization of the strip using granularities g_x and g_y . The authors highlight that these values directly influence the quality of the solution and the number of variables and constraints of the problem.

The indices used in the model are presented as follows:

- $t, u \in \mathcal{T} = \{1, \dots, T\}$ - are piece types;
- $c \in \mathcal{C} = \{0, \dots, C - 1\}$ - is a column of the grid;
- $r \in \mathcal{R} = \{0, \dots, R - 1\}$ - is a row of the grid;
- $d, e \in \mathcal{D} = \{1, \dots, D\}$ - are dots of the grid. Alternatively, each dot can be represented by an ordered pair of values $(c, r) \in \mathcal{C} \times \mathcal{R}$, such that $c = \lfloor \frac{d}{R} \rfloor$ and $r = d - \lfloor \frac{d}{R} \rfloor (R - 1)$. Analogously, $d = c R - r + 1$.

In this model, each piece type t is represented by a polygon and it has a quantity of pieces to be cut q_t . The polygon is described by a set of vertices whose coordinates are defined from a

reference point. Furthermore, each piece type t has a rectangular envelope that define, respectively, lower bounds x_t^m and y_t^m and upper bounds x_t^M and y_t^M for axes x and y , as illustrated by Figure 6.

Figure 3 – Representation of a piece type.

Source: Elaborated by the author.

First, to impose each piece is entirely inside the strip, we use the inner-fit polygon. Given a piece type t and the strip, the inner-fit polygon $\mathcal{IFP}_t$ is the set of dots of the strip where the reference point of a piece of type t can be place without extrapolate the boards of the strip (Figure 7).

Figure 4 – Example of inner-fit polygon in the grid.

Source: Elaborated by the author.

To impose there are no overlap between two pieces we use the nofit polygon. Given two piece types t and u and a dot d , the nofit polygon $\mathcal{NFP}_{t,u}^d$ is the set of dots that are inside of $\mathcal{NFP}_{t,u}$ if a pice of type t is place in dot d (Figure 8). Note that pice touch each other in the black dots and overlap in the white ones.

After the introduction of the inner-fit polygon and nofit polygon concepts, we need to describe the variables used in the model. For each piece type t and dot of the strip d , such that $d \in \mathcal{IFP}_t$, we have a binary decision variable δ_t^d , defined as follows.

$$\delta_t^d = \begin{cases} 1, & \text{if a piece of type } t \text{ is place in dot } d \text{ of the strip;} \\ 0, & \text{otherwise.} \end{cases} \quad (2.1)$$

The model proposed by Toledo *et al.* (2013) is given by:

Figure 5 – Construction of the nofit polygon in the grid.

Source: Elaborated by the author.

$$\min \quad z \quad (2.2)$$

$$\text{s.a:} \quad (c_d g_x + x_t^M) \delta_t^d \leq z, \quad \forall t \in \mathcal{T}, \forall d \in \mathcal{IFP}_t, \quad (2.3)$$

$$\sum_{d \in \mathcal{IFP}_t} \delta_t^d = q_t, \quad \forall t \in \mathcal{T}, \quad (2.4)$$

$$\delta_t^d + \delta_u^e \leq 1, \quad \forall t, u \in \mathcal{T}, t \leq u, \forall d \in \mathcal{IFP}_t, \forall e \in \mathcal{NFP}_{t,u}^d, \quad (2.5)$$

$$\delta_t^d \in \{0, 1\}, \quad \forall t \in \mathcal{T}, \forall d \in \mathcal{IFP}_t, \quad (2.6)$$

$$z \geq 0, \quad (2.7)$$

Objective function (2.8) together with constraints (2.9) minimize the total length of the strip needed to cut all the pieces. Constraints (2.10) imposes that all pieces are placed completely inside the strip, while constraints (2.11) prevent overlap among them. Constraints (2.12) and (2.13) define, respectively, the domain of binary variables and the variable associated to the length of the strip.

2.2 Problem definition

The irregular strip packing problem can be formally described as follows. Let $\mathcal{P} = \{1, \dots, P\}$ be a set of two-dimensional pieces (not necessarily distinct) and W be the width of an infinite length strip. Consider each piece $i \in \mathcal{P}$ is described by a polygon with an associated reference point $v_i = (x_i, y_i)$. From this point, the piece position is defined on the strip. The problem consists in defining a cutting or packing layout that disposes all pieces $i \in \mathcal{P}$ inside the strip wObjective function (2.8) together with constraints (2.9) minimize the total length of the strip needed to cut all the pieces. Constraints (2.10) imposes that all pieces are placed completely inside the strip, while constraints (2.11) prevent overlap among them. Constraints (2.12) and (2.13) define, respectively, the domain of binary variables and the variable associated to the

length of the strip without overlap, in such a way strip length z is minimum. Here, no piece rotation was considered.

This study is based on the dotted-board model (TOLEDO *et al.*, 2013), a mixed integer programming model for the irregular strip packing problem, in which the strip is represented by a grid. Toledo *et al.* calculated an upper bound (L) for the strip length and applied a discretization using an appropriate resolution g_x (horizontal) by g_y (vertical). Such values directly influence the solution quality, since a refined discretization increases the number of constraints and binary variables used.

The following are the indices used in the model:

- $t, u \in \mathcal{T} = \{1, \dots, T\}$ - are piece types;
- $c \in \mathcal{C} = \{0, \dots, C-1\}$ - is a board column;
- $r \in \mathcal{R} = \{0, \dots, R-1\}$ - is a board row;
- $d, e \in \mathcal{D} = \{1, \dots, D\}$ - are board dots.

In the model, each piece type $t \in \mathcal{T}$ is represented by a polygon and has a demand q_t . The polygon is described by a set of vertices whose coordinates are defined from a reference point. Furthermore, each piece type t has a rectangular enclosure that defines, respectively, lower bounds x_t^m and y_t^m and upper bounds x_t^M and y_t^M for x and y axes, as shown in Figure 6.

Figure 6 – Representation of a piece type.

Source: Elaborated by the author.

The inner-fit polygon concept is used to ensure each piece is entirely inside the board. Given a piece type t , the inner-fit polygon $\mathcal{IFP}_t$ is the set of dots where the reference point of a piece of type t imposes a piece stays inside the board (Figure 7). The column of dot $d \in \mathcal{IFP}_t$ is denoted by c_d and its row is represented by r_d .

For each piece type $t \in \mathcal{T}$ and dot d from the board, such that $d \in \mathcal{IFP}_t$, there is a binary decision variable δ_t^d , defined as follows.

$$\delta_t^d = \begin{cases} 1, & \text{if a piece of type } t \text{ is allocated in dot } d \text{ in the board;} \\ 0, & \text{otherwise.} \end{cases}$$

Figure 7 – Example of an inner-fit polygon on the grid.

Source: Elaborated by the author.

Toledo *et al.* (2013) used the nofit polygon concept to ensure two pieces do not overlap. Given two pieces t and u , the nofit polygon $NFP_{t,u}$ is the region where the reference point of piece u can not be placed, since it would overlap piece t . $NFP_{t,u}$ can be geometrically interpreted as the region defined by the trajectory followed by the reference point of piece u when it slides around another piece t and vice-versa (Figure 8 (a)). Similarly, the nofit polygon $\mathcal{NFP}_{t,u}^d$ is the set of dots inside $NFP_{t,u}$ if a piece of type t is positioned in dot d (Figure 8 (b)). Note the two pieces touch each other in the black dots and overlap in the white ones.

Figure 8 – Construction of a nofit polygon on the grid.

Source: Elaborated by the author.

The dotted-board model is given by

$$\min \quad z \quad (2.8)$$

$$\text{s.t.} \quad (c_d g_x + x_t^M) \delta_t^d \leq z, \quad \forall t \in \mathcal{T}, \forall d \in \mathcal{IFP}_t, \quad (2.9)$$

$$\sum_{d \in \mathcal{IFP}_t} \delta_t^d = q_t, \quad \forall t \in \mathcal{T}, \quad (2.10)$$

$$\delta_t^d + \delta_u^e \leq 1, \quad \forall t, u \in \mathcal{T}, \forall d \in \mathcal{IFP}_t, \forall e \in \mathcal{NFP}_{t,u}^d, \quad (2.11)$$

$$\delta_t^d \in \{0, 1\}, \quad \forall t \in \mathcal{T}, \forall d \in \mathcal{IFP}_t, \quad (2.12)$$

$$z \geq 0, \quad (2.13)$$

First, objective function (2.8) together with constraints (2.9) minimize the length of the board used to cut all pieces. Constraints (2.10) ensure the demand is satisfied for all pieces, while constraints (2.11) prevent overlap between pieces. Finally, constraints (2.12) and (2.13) define the variable domains.

2.3 Clique covering model

This section describes the clique covering model. In Subsection 3.1, the non-overlap constraints are represented by an edge clique covering and in Subsection 3.2, we define the objective function based on a vertex clique covering. Subsection 3.3 presents the proposed model and Subsection 3.4 introduces valid inequalities and a lower bound for the proposed model.

2.3.1 Non-overlap constraints

For the dotted-board model, the overlap between pieces can be described by an undirected graph $G = (V, E)$, where V is the set of vertices representing binary variables (ordered pair consisting of a piece type and a board dot) and E is the set of edges representing non-overlap constraints. Therefore, for each piece type $t \in \mathcal{T}$ and dot $d \in \mathcal{IFP}_t$, there is a corresponding binary variable δ_t^d described by a vertex $v = (d, t) \in V$. Furthermore, given two vertices $v_1 = (d, t)$ and $v_2 = (e, u)$, such that $e \in \mathcal{NFP}_{t,u}^d$, i.e., the set of dots where the pieces overlap if one piece of type t is positioned in dot $d \in \mathcal{IFP}_t$ and another piece of type u is positioned in dot $e \in \mathcal{IFP}_u$, we describe this overlap by an edge $\{v_1, v_2\}$.

Figure 9 illustrates an irregular strip packing instance. Figures 9 (a) and 9 (b) show, respectively, two piece types and their reference points and the admissible points on the grid for each piece, $\mathcal{IFP}_1 = \{1, 2, 4, 5\}$ and $\mathcal{IFP}_2 = \{2, 3\}$. Figure 9 (c) shows a feasible cutting layout for this instance. Given the nofit polygons between the piece types (Figure 10), the overlap is represented by a conflict graph (Figure 11), where same color vertices represent the same piece type. If the demand is one for each piece type, a feasible solution to this problem can be obtained from two non-adjacent vertices with distinct colors on the graph. Therefore, the vertices associated with the variables $\{\delta_1^1, \delta_2^3\}$ describe a feasible solution to the problem (Figure 9 (c)).

We can generalize the previous example with an arbitrary number of irregular pieces. Basically, a feasible solution on the grid can be obtained through a stable set that satisfies demand constraints (2.10). We can replace non-overlap constraints (2.11) by the following constraints, which represent an edge clique covering $\mathcal{E}$:

$$\sum_{(d,t) \in K} \delta_t^d \leq 1, \quad \forall K \in \mathcal{E}. \quad (2.14)$$

The replacement of edge constraints (2.11) by cliques (2.14) leads to many advantages. First, for the studied problem, the number of constraints used to represent overlap between pieces

Figure 9 – Example of an irregular strip packing problem. (a) Piece types. (b) Admissible points. (c) Feasible solution.

Source: Elaborated by the author.

Figure 10 – Nofit polygons for pieces 1 and 2.

Source: Elaborated by the author.

Figure 11 – Example of a conflict graph.

Source: Elaborated by the author.

is drastically reduced. Second, clique constraints also reduce the number of nonzeros of the model, which favors methods for sparse matrices. Last, cliques can significantly improve the lower bound obtained from the linear relaxation of the model.

The edge clique covering problem regards the determination of the minimum number of cliques that cover all edges of a graph G , i.e., each edge must be covered by at least one clique. If each edge must be covered by exactly one clique, the problem is called edge clique partitioning. The minimum number of cliques required to cover all the edges of graph G (clique

covering number) is denoted by $cc(G)$, while the minimum number required to partition (clique partitioning number) is denoted by $cp(G)$. Note $cc(G) \leq cp(G)$.

The edge clique covering problem is NP-hard (ORLIN, 1977; KOU; STOCKMEYER; WONG, 1978) and equivalent to the keyword conflict problem (KOU; STOCKMEYER; WONG, 1978) and the intersection graph problem (ERDŐS; GOODMAN; PÓSA, 1966). Two reviews on the edge clique covering problem can be found in Pullman (1983) and Monson, Pullman and Rees (1995). More information about the applications of this problem is provided by Roberts (1985).

Here, we use the Maximum 1 Expand Kou (MAX1_EK) heuristic proposed by Rodrigues and Miyazawa (2018) to solve the edge clique covering problem. They generated an initial clique covering, expanded each clique using a maximal clique generation algorithm and used the post-processing method proposed by Kou, Stockmeyer and Wong (1978) to eliminate redundant cliques.

2.3.2 Objective function

To calculate the objective function, we must find the highest coefficient associated with active binary variables. Instead of using constraints (2.9) for each vertex, we use a vertex clique covering $\mathcal{V}$ given by the following constraints:

$$\sum_{(d,t) \in K} (c_d g_x + x_t^M) \delta_t^d \leq z, \quad \forall K \in \mathcal{V}. \quad (2.15)$$

The replacement of vertex constraints (2.9) by cliques (2.15) leads to the same advantages described in the previous subsection, i.e., the number of constraints and nonzeros is reduced and the lower bound of the model can be improved.

The vertex clique covering problem regards the determination of the minimum number of cliques that cover all vertices of a graph G , i.e., each vertex must be covered by at least one clique. If each vertex must be covered by only one clique, the problem is called a vertex clique partitioning. Note this problem is equivalent to the graph coloring problem on complement graph $\overline{G}$.

The graph coloring problem is NP-hard (KARP, 1972). For surveys on graph coloring, see Pardalos, Mavridou and Xue (1998) and Malaguti and Toth (2010) and for a specific review on local optimization methods, see Galinier and Hertz (2006).

We used the Recursive Largest First (RLF) heuristic proposed by Leighton (1979) for the resolution of the graph coloring problem on the complement graph (vertex clique covering) and lifted the cliques using a maximal clique generation algorithm (RLFM heuristic).

2.3.3 Proposed Model

Let $K \subseteq V$ be a clique from G and $\mathcal{V}$ and $\mathcal{E}$ be a vertex and an edge clique covering of G . The clique covering model is given by

$$\min \quad z \quad (2.16)$$

$$\text{s.a:} \quad \sum_{(d,t) \in K} (c_d g_x + x_t^M) \delta_t^d \leq z, \quad \forall K \in \mathcal{V}, \quad (2.17)$$

$$\sum_{d \in \mathcal{IFP}_t} \delta_t^d = q_t, \quad \forall t \in \mathcal{T}, \quad (2.18)$$

$$\sum_{(d,t) \in K} \delta_t^d \leq 1, \quad \forall K \in \mathcal{E}, \quad (2.19)$$

$$\delta_t^d \in \{0, 1\}, \quad \forall t \in \mathcal{T}, \forall d \in \mathcal{IFP}_t. \quad (2.20)$$

Objective function (3.2) and constraints (4.19) and (2.20) are the same of the dotted-board model. Vertex clique covering constraints (2.17) define the length of the board used to cut all pieces, while edge clique covering constraints (2.19) impose non-overlap between pieces. Note the dotted-board model is a particular case of the clique covering model if the trivial coverings are used for $\mathcal{V}$ and $\mathcal{E}$.

2.3.4 Valid inequalities and lower bound

In the dotted-board model, we consider auxiliary constraints (2.9) and demand constraints (2.10) simultaneously

$$(c_d g_x + x_t^M) \delta_t^d \leq z, \quad \forall t \in \mathcal{T}, \forall d \in \mathcal{IFP}_t, \quad (2.9)$$

$$\sum_{d \in \mathcal{IFP}_t} \delta_t^d = q_t, \quad \forall t \in \mathcal{T}, \quad (2.10)$$

to generate the following valid inequalities:

$$\sum_{d \in \mathcal{IFP}_t} (c_d g_x + x_t^M) \delta_t^d \leq q_t z, \quad \forall t \in \mathcal{T}. \quad (2.21)$$

Let $\underline{z}$ be a lower bound for the problem. It follows that

$$z \geq \underline{z} \geq \underline{z} \delta_t^d, \quad \forall t \in \mathcal{T}, \forall d \in \mathcal{IFP}_t. \quad (2.22)$$

We can lift the coefficients from inequalities (2.21) using inequalities (2.22) and the following constraints are obtained:

$$\sum_{d \in \mathcal{IFP}_t} \max\{\underline{z}, (c_d g_x + x_t^M)\} \delta_t^d \leq q_t z, \quad \forall t \in \mathcal{T}. \quad (2.23)$$

Valid inequalities (2.23) aggregate plenty of information, such as demand, coefficients associated with the objective function and lower bound value of the problem. Although such inequalities can improve the value obtained by the linear relaxation of the model, they may hamper its resolution.

Similarly, we derive valid inequalities for any clique obtained from the conflict graph of the clique covering model including the trivial cliques of the dotted-board model:

$$\sum_{(d,t) \in K} \max\{\underline{z}, (c_d g_x + x_t^M)\} \delta_t^d \leq z, \quad \forall K \in \mathcal{V}. \quad (2.24)$$

$$\sum_{(d,t) \in K} \max\{\underline{z}, (c_d g_x + x_t^M)\} \delta_t^d \leq z, \quad \forall K \in \mathcal{E}, \quad (2.25)$$

Since $\underline{z}$ is a lower bound for the problem, i.e. $z \geq \underline{z}$, we can ignore all variables δ_t^d whose objective function coefficient is lower than or equal to $\underline{z}$ to calculate board length z . Let $G(\underline{z}^+)$ be the subgraph of G induced by the vertices whose coefficient is greater than $\underline{z}$. Let $\mathcal{V}_{\underline{z}^+}$ and $\mathcal{E}_{\underline{z}^+}$ be, respectively, a vertex and an edge clique covering of $G(\underline{z}^+)$. We can derive the following valid inequalities:

$$q_t \underline{z} + \sum_{(d,t) \in G(\underline{z}^+)} (c_d g_x + x_t^M - \underline{z}) \delta_t^d \leq q_t z, \quad \forall t \in \mathcal{T}. \quad (2.26)$$

$$\underline{z} + \sum_{(d,t) \in K} (c_d g_x + x_t^M - \underline{z}) \delta_t^d \leq z, \quad \forall K \in \mathcal{V}_{\underline{z}^+}. \quad (2.27)$$

$$\underline{z} + \sum_{(d,t) \in K} (c_d g_x + x_t^M - \underline{z}) \delta_t^d \leq z, \quad \forall K \in \mathcal{E}_{\underline{z}^+}. \quad (2.28)$$

Constraints (2.27) can replace constraints (2.15) in the clique covering model. As the dotted-board model is a particular case of the clique covering model, constraints (2.27) can also replace constraints (2.9).

We highlight valid inequalities (2.23) - (2.28) can be used for the construction of an iterative method that improves the lower bound of the problem (Algorithm 1). In each iteration, we solve a linear relaxation of the clique covering model through the addition of valid inequalities. If the stopping criteria have not been reached, new valid inequalities can be generated based on the new lower bound value.

2.4 Computational experiments

This section presents the evaluated instances and the results of the clique covering and dotted-board models. All algorithms described in the previous sections were coded in C++ with

Algorithm 1 – Iterative method for the calculation of a lower bound.

input : Clique covering model $\mathcal{M}$;

 Lower bound $\underline{z}$.

output : new lower bound $\underline{z}'$.

while stopping criteria are not satisfied **do**

 Add valid inequalities from (2.23) - (2.28) to model $\mathcal{M}$ $\underline{z}' \leftarrow$ linear relaxation of model $\mathcal{M}$
 // Lower bound update

if $\underline{z} < \underline{z}'$ **then**

 | $\underline{z} \leftarrow \underline{z}'$
end
end
return $\underline{z}'$

Standart Template Library (STL) data structures. The experiments were run on a PC with Intel® Core™ i7-2600 @3.4 GHz processor and 16 GB RAM under Ubuntu 14.04 LTS 64 bits system. A discretization of one unit ($g_x = g_y = 1$) was used for all instances solved by the commercial MIP solver IBM ILOG CPLEX 12.6 with default parameters. We used two time limits of 18,000 s and 3,600 s to evaluate both models.

We compared both models using 81 instances taken from the literature. The instances consist of 12 groups: three (ALVAREZ-VALDES; MARTINEZ; TAMARIT, 2013), shapes (OLIVEIRA; FERREIRA, 1993), BLAZEWCZ (BŁAŻEWICZ; HAWRYLUK; WALKOWIAK, 1993), RCO (RIBEIRO; CARRAVILLA; OLIVEIRA, 1999), artif (DOWSLAND; DOWSLAND; BENNELL, 1998), shirts (DOWSLAND; DOWSLAND; BENNELL, 1998), dagli (RATANAPAN; DAGLI, 1997), fu (FUJITA; AKAGI; HIROKAWA, 1993), J1 (ALVAREZ-VALDES; MARTINEZ; TAMARIT, 2013), J2 (ALVAREZ-VALDES; MARTINEZ; TAMARIT, 2013), poly (HOPPER, 2000) and jakobs (JAKOBS, 1996). They are well-known benchmark instances and most of them are available at EURO Special Interest Group on Cutting and Packing (ES-ICUP) website ¹. We also introduced some instances based on a subset of literature instances. Poly1b/c/d/e instances are groups of 15 consecutive pieces (16-30/31-45/46-60/61-75) taken from poly5b. To generate artifa_b and shirtsa_b instances, we took a and b copies from pieces whose demands are, respectively, eight and 15, in the artif and shirts instances. Dagli1 instance consists of only one copy for each piece from instance dagli.

The characteristics of the evaluated instances are shown in Table 1. The first column provides the instance name, whereas the second and third columns describe the number of pieces and piece types for each instance. The fourth and fifth columns show the board width and length. This length is an upper bound obtained by a bottom-left-fill heuristic subject to the defined discretization. We ran the algorithm using eight sorting criteria described by Dowsland, Vaid and Dowsland (2002) and selected the best solution. The last two columns report the number of vertices and edges of the conflict graph obtained from the nofit polygon.

¹ <<https://paginas.fe.up.pt/~esicup/datasets>>

Table 1 – Features of data instances.

Instance	Pieces		Board		Graph	
	Total	Types	Width	Length	Vertices	Edges
three	3	3	7	7	61	1,266
threep2	6	3	7	11	117	3,006
threep2w9	6	3	9	9	127	3,394
threep3	9	3	7	16	187	5,181
threep3w9	9	3	9	13	207	6,150
shapes4	4	4	13	24	389	46,060
shapes8	8	4	20	28	1,173	250,050
SHAPES2	8	4	40	16	848	92,766
SHAPES4	16	4	40	28	2,468	604,259
SHAPES5	20	4	40	35	3,413	939,916
SHAPES7	28	4	40	48	5,168	1,563,279
SHAPES9	34	4	40	54	5,978	1,850,985
SHAPES15	43	4	40	67	7,733	2,474,348
blasz2	20	4	15	27	1,080	91,838
BLAZEWICZ1	7	7	15	8	432	35,478
BLAZEWICZ2	14	7	15	16	1,096	120,894
BLAZEWICZ3	21	7	15	22	1,594	184,956
BLAZEWICZ4	28	7	15	29	2,175	259,695
BLAZEWICZ5	35	7	15	36	2,756	334,434
RCO1	7	7	15	8	432	36,049
RCO2	14	7	15	17	1,179	133,924
RCO3	21	7	15	25	1,843	220,924
RCO4	28	7	15	29	2,175	264,424
RCO5	35	7	15	41	3,171	394,924
artif1_2	13	8	27	8	947	68,609
artif2_4	26	8	27	14	2,099	250,427
artif3_6	39	8	27	20	3,251	437,573
artif4_8	52	8	27	28	4,787	687,101
artif5_10	65	8	27	34	5,939	874,247
artif6_12	78	8	27	41	7,283	1,092,584
artif7_14	91	8	27	48	8,627	1,310,921
artif	99	8	27	53	9,587	1,466,876
shirts1_2	13	8	40	13	1,924	147,101
shirts2_4	26	8	40	20	4,010	736,227
shirts3_6	39	8	40	26	5,798	1,364,984
shirts4_8	52	8	40	35	8,480	2,319,578
shirts5_10	65	8	40	42	10,566	3,062,040
dagli1	10	10	60	25	6,899	6,204,530
fu5	5	4	38	18	721	120,663
fu6	6	5	38	24	1,722	734,345
fu7	7	6	38	24	2,172	1,124,614
fu8	8	7	38	24	2,772	1,616,281
fu9	9	8	38	29	4,202	3,446,231
fu10	10	9	38	34	5,982	6,365,641
fu	12	11	38	38	8,642	11,355,356
J1-10-10-0	10	9	10	20	1,063	118,778
J1-10-10-1	10	10	10	19	1,065	154,905
J1-10-10-2	10	9	10	21	1,073	140,430
J1-10-10-3	10	10	10	23	1,284	232,436
J1-10-10-4	10	10	10	15	896	96,289
J1-12-20-0	12	11	20	12	1,597	260,436
J1-12-20-1	12	11	20	11	1,377	207,624
J1-12-20-2	12	12	20	14	2,039	445,512
J1-12-20-3	12	12	20	10	1,363	172,873
J1-12-20-4	12	12	20	16	2,408	580,191
J1-14-20-0	14	13	20	14	2,344	469,901
J1-14-20-1	14	13	20	14	2,319	459,468
J1-14-20-2	14	14	20	16	2,781	776,783
J1-14-20-3	14	13	20	12	1,979	348,312
J1-14-20-4	14	13	20	16	2,631	634,924
J2-10-35-0	10	9	35	28	4,486	3,505,514
J2-10-35-1	10	9	35	28	4,572	3,569,796
J2-10-35-2	10	10	35	27	4,892	3,811,514
J2-10-35-3	10	9	35	25	4,328	2,875,373
J2-10-35-4	10	10	35	22	3,992	2,681,176
J2-12-35-0	12	11	35	31	6,240	6,493,882
J2-12-35-1	12	11	35	29	6,100	5,234,224
J2-12-35-2	12	10	35	30	5,852	4,829,349
J2-12-35-3	12	11	35	25	5,420	4,150,239
J2-12-35-4	12	11	35	29	6,256	5,439,275
J2-14-35-0	14	13	35	34	8,718	10,161,001
J2-14-35-1	14	13	35	33	8,620	9,309,051
J2-14-35-2	14	11	35	33	7,484	6,818,098
J2-14-35-3	14	13	35	29	7,552	7,561,900
J2-14-35-4	14	13	35	31	8,056	8,157,073
poly1a	15	15	40	18	5,620	3,285,681
poly1b	15	15	40	21	6,858	4,922,991
poly1c	15	15	40	14	4,437	1,513,472
poly1d	15	15	40	14	4,527	1,947,175
poly1e	15	15	40	13	3,638	1,301,446
jakobs1	25	22	40	13	7,460	2,846,369

After the construction of the conflict graph, we generated a vertex clique covering by RLFM algorithm (LEIGHTON, 1979) and an edge clique covering by MAX1_EK algorithm (RODRIGUES; MIYAZAWA, 2018). RLFM ran in less than 20 s and MAX1_EK in less than 150 s for all instances.

We used the iterative method described in Algorithm 1 to improve the model lower bound. The inputs of the algorithm were the model (3.2) - (2.20) with valid inequalities (2.26) and (2.27) and the trivial lower bound defined as the maximum between the maximum piece length and the sum of the area of all pieces divided by the board width, i.e.:

$$\underline{z} = \max \left\{ \max_{t \in \mathcal{T}} (x_t^M - x_t^m), \frac{1}{W} \sum_{t \in \mathcal{T}} q_t A_t \right\}. \quad (2.29)$$

We solved the linear relaxation in each iteration and if the objective function increased, only constraints (2.26) were updated. The stop criterion is the objective function difference $|z' - \underline{z}| \leq 10^{-6}$. If the lower bound is fractional, we apply the ceiling function, since we use the integer discretization and all evaluated instances have polygons with integer lengths.

Table 2 shows the results of the iterative method. The first column shows the instance name, whereas the trivial lower bound (LB^T) and the iterative lower bound (LB^I) are described in second and third columns. The fourth column highlights the percentage of lower bound improvement ($LB^{IMP} = 100 \frac{LB^I - LB^T}{LB^T}$) obtained by the proposed method. The fifth and sixth columns provide the execution time (in seconds) and number of iterations, respectively. The average, minimum and maximum values for all instances are summarized in the three bottom rows.

For most instances, the lower bound obtained by the iterative method was better than the trivial one, increasing 16.12% the lower bound value on average. The method converged in less than five minutes and performed few iterations for all instances. However, there was no improvement for nine instances, mainly when the maximum piece length was greater than the area bound (instances SHAPES2, artif1_2, dagli1, poly1c and poly1d).

We used the improved lower bound to add valid inequalities to both clique covering and dotted-board models. In fact, we replaced constraints (2.9) or (2.15) by (2.27). Table 3 shows the number of binaries, constraints and nonzeros of the clique covering and dotted-board models. For each model we also report the number of objective function and non-overlap constraints.

As shown in Table 3, the number of constraints of the dotted-board model was significantly reduced by the coverings. For example, for instance fu 11,357,565 constraints were replaced by only 5,228. The number of nonzeros was also reduced from 22,723,750 to 3,925,065.

The computational results of the clique covering and dotted-board models within time limits of 18,000 s and 3,600 s are shown in Table 4 and 5. For each model and instance, the tables report upper bound (UB) and lower bound (LB), optimality gap ($GAP = 100 \frac{UB - LB}{UB}$),

Table 2 – Trivial and iterative lower bounds.

Instance	Trivial LB	Iterative LB	Improvement (%)	Time (s)	Iteration
three	4	6	50.00	0.00	3
threep2	7	10	42.86	0.00	3
threep2w9	6	8	33.33	0.01	3
threep3	10	14	40.00	0.01	4
threep3w9	8	11	37.50	0.01	3
shapes4	14	23	64.29	0.16	6
shapes8	16	23	43.75	7.57	5
SHAPES2	14	14	0.00	0.08	1
SHAPES4	16	21	31.25	127.73	4
SHAPES5	20	25	25.00	28.49	5
SHAPES7	28	34	21.43	108.49	6
SHAPES9	33	38	15.15	133.25	5
SHAPES15	40	47	17.50	281.59	7
blasz2	19	24	26.32	1.00	5
BLAZEWICZ1	6	8	33.33	0.07	3
BLAZEWICZ2	11	13	18.18	1.46	3
BLAZEWICZ3	17	19	11.76	5.69	3
BLAZEWICZ4	22	25	13.64	18.41	4
BLAZEWICZ5	27	31	14.81	48.46	5
RCO1	7	8	14.29	0.02	2
RCO2	13	14	7.69	0.33	2
RCO3	19	21	10.53	1.72	3
RCO4	26	27	3.85	2.26	2
RCO5	32	34	6.25	0.97	3
artif1_2	7	7	0.00	0.02	1
artif2_4	10	12	20.00	9.57	3
artif3_6	14	17	21.43	6.35	4
artif4_8	19	22	15.79	14.74	4
artif5_10	23	27	17.39	25.49	4
artif6_12	28	33	17.86	41.78	5
artif7_14	33	38	15.15	65.05	5
artif	36	41	13.89	96.36	5
shirts1_2	13	13	0.00	0.01	1
shirts2_4	14	17	21.43	3.50	3
shirts3_6	21	24	14.29	14.47	3
shirts4_8	28	31	10.71	58.80	4
shirts5_10	35	37	5.71	80.54	3
dagli1	23	23	0.00	32.07	1
fu5	14	18	28.57	0.04	2
fu6	17	23	35.29	3.80	6
fu7	19	24	26.32	4.90	5
fu8	20	24	20.00	15.90	4
fu9	23	24	4.35	10.16	2
fu10	26	28	7.69	21.58	3
fu	29	31	6.90	182.71	3
J1-10-10-0	16	18	12.50	0.27	3
J1-10-10-1	16	16	0.00	0.14	1
J1-10-10-2	18	19	5.56	0.16	2
J1-10-10-3	18	20	11.11	0.82	3
J1-10-10-4	11	12	9.09	0.08	2
J1-12-20-0	10	11	10.00	0.53	2
J1-12-20-1	9	10	11.11	0.53	2
J1-12-20-2	11	12	9.09	0.72	2
J1-12-20-3	7	8	14.29	0.34	2
J1-12-20-4	12	12	0.00	0.47	1
J1-14-20-0	11	12	9.09	1.00	2
J1-14-20-1	11	11	0.00	0.21	1
J1-14-20-2	13	13	0.00	0.65	1
J1-14-20-3	9	10	11.11	0.48	2
J1-14-20-4	12	13	8.33	5.66	2
J2-10-35-0	19	22	15.79	102.08	3
J2-10-35-1	16	22	37.50	107.56	5
J2-10-35-2	17	21	23.53	76.72	4
J2-10-35-3	16	18	12.50	29.21	3
J2-10-35-4	14	17	21.43	34.77	3
J2-12-35-0	22	25	13.64	249.24	4
J2-12-35-1	18	23	27.78	179.60	5
J2-12-35-2	18	21	16.67	148.62	4
J2-12-35-3	16	19	18.75	75.31	4
J2-12-35-4	20	24	20.00	114.18	5
J2-14-35-0	24	26	8.33	254.99	3
J2-14-35-1	22	25	13.64	238.19	4
J2-14-35-2	21	23	9.52	145.02	3
J2-14-35-3	21	22	4.76	112.94	2
J2-14-35-4	21	24	14.29	191.63	4
poly1a	13	14	7.69	24.36	2
poly1b	13	16	23.08	71.23	3
poly1c	13	13	0.00	2.72	1
poly1d	11	11	0.00	3.93	1
poly1e	10	11	10.00	5.09	2
jakobs1	10	11	10.00	2.36	2
AVG	17.36	19.96	16.12	44.91	3.16
MAX	40	47	64.29	281.59	7
MIN	4	6	0.00	0.00	1

Table 3 – Statistics of the clique covering and dotted-board models.

Instance	Binaries	Clique covering model			Nonzeros	Dotted-board model			Nonzeros
		Constraints				Constraints			
		Obj.	Overlap	Total		Obj.	Overlap	Total	
three	61	2	24	29	615	14	1,266	1,283	2,621
threep2	117	2	64	69	1,568	14	3,006	3,023	6,157
threep2w9	127	3	72	78	1,743	20	3,394	3,417	6,955
threep3	187	2	115	120	2,777	28	5,181	5,212	10,605
threep3w9	207	3	140	146	3,366	40	6,150	6,193	12,587
shapes4	389	2	294	300	23,459	27	46,060	46,091	92,563
shapes8	1,173	10	1,079	1,093	120,369	440	250,050	250,494	502,153
SHAPES2	848	9	778	791	58,945	270	92,766	93,040	186,920
SHAPES4	2,468	19	2,771	2,794	330,726	945	604,259	605,208	1,212,876
SHAPES5	3,413	27	3,824	3,855	482,873	1,350	939,916	941,270	1,885,945
SHAPES7	5,168	36	5,885	5,925	777,144	1,890	1,563,279	1,565,173	3,135,506
SHAPES9	5,978	40	6,751	6,795	901,127	2,160	1,850,985	1,853,149	3,712,268
SHAPES15	7,733	50	8,682	8,736	1,180,592	2,700	2,474,348	2,477,052	4,961,829
blasz2	1,080	3	707	714	43,042	135	91,838	91,977	185,026
BLAZEWCZ1	432	0	258	265	16,487	0	35,478	35,485	71,388
BLAZEWCZ2	1,096	14	723	744	52,964	249	120,894	121,150	243,382
BLAZEWCZ3	1,594	14	1,067	1,088	79,494	249	184,956	185,212	372,004
BLAZEWCZ4	2,175	14	1,472	1,493	110,255	332	259,695	260,034	522,229
BLAZEWCZ5	2,756	21	1,884	1,912	142,442	415	334,434	334,856	672,454
RCO1	432	0	149	156	10,594	0	36,049	36,056	72,530
RCO2	1,179	14	454	475	37,631	249	133,924	134,180	269,525
RCO3	1,843	14	705	726	59,320	332	220,924	221,263	444,355
RCO4	2,175	7	829	843	69,400	166	264,424	264,597	531,355
RCO5	3,171	28	1,175	1,210	100,540	581	394,924	395,512	794,181
artif1_2	947	18	456	482	22,107	192	68,609	68,809	138,549
artif2_4	2,099	25	1,423	1,456	107,268	384	250,427	250,819	503,721
artif3_6	3,251	33	2,388	2,429	194,039	576	437,573	438,157	879,549
artif4_8	4,787	59	3,777	3,844	320,027	1,152	687,101	688,261	1,381,293
artif5_10	5,939	66	4,782	4,856	410,619	1,344	874,247	875,599	1,757,121
artif6_12	7,283	76	5,958	6,042	516,297	1,536	1,092,584	1,094,128	2,195,523
artif7_14	8,627	92	7,163	7,263	625,446	1,920	1,310,921	1,312,849	2,634,309
artif	9,587	107	7,969	8,084	699,310	2,304	1,466,876	1,469,188	2,947,947
shirts1_2	1,924	0	569	577	42,955	0	147,101	147,109	296,126
shirts2_4	4,010	40	1,681	1,729	263,506	894	736,227	737,129	1,478,252
shirts3_6	5,798	40	2,892	2,940	526,306	596	1,364,984	1,365,588	2,736,958
shirts4_8	8,480	40	4,766	4,854	943,501	1,192	2,319,578	2,320,778	4,650,020
shirts5_10	10,566	80	6,209	6,297	1,263,251	1,490	3,062,040	3,063,538	6,137,626
dagli1	6,899	20	4,630	4,660	2,656,790	982	157,661	6,205,522	12,417,923
fu5	721	0	193	197	35,383	0	120,663	120,667	242,047
fu6	1,722	6	579	590	229,071	146	734,345	734,496	1,470,704
fu7	2,172	0	726	732	336,890	0	1,124,614	1,124,620	2,251,400
fu8	2,772	0	1,083	1,090	469,725	0	1,616,281	1,616,288	3,235,334
fu9	4,202	9	1,721	1,738	886,064	1,155	3,446,231	3,447,394	6,898,974
fu10	5,982	12	2,217	2,238	1,389,530	1,536	6,365,641	6,367,186	12,740,336
fu	8,642	14	5,203	5,228	3,925,065	2,198	11,355,356	11,357,565	22,723,750
J1-10-10-0	1,063	8	400	417	30,092	126	118,778	118,913	238,871
J1-10-10-1	1,065	11	548	569	52,784	210	154,905	155,125	311,295
J1-10-10-2	1,073	7	367	383	32,492	122	140,430	140,561	282,177
J1-10-10-3	1,284	7	472	489	58,460	207	232,436	232,653	466,570
J1-10-10-4	896	12	261	283	21,005	228	96,289	96,527	193,930
J1-12-20-0	1,597	14	498	522	52,793	185	260,436	260,632	522,839
J1-12-20-1	1,377	10	611	632	61,710	188	207,624	207,823	417,001
J1-12-20-2	2,039	15	590	617	76,736	398	445,512	445,922	893,859
J1-12-20-3	1,363	26	406	433	36,852	422	172,873	173,212	347,744
J1-12-20-4	2,408	31	726	769	112,145	784	580,191	580,987	1,164,358
J1-14-20-0	2,344	19	617	649	77,751	442	469,901	470,356	943,030
J1-14-20-1	2,319	20	874	907	112,225	672	459,468	460,153	922,599
J1-14-20-2	2,781	22	718	754	124,506	705	776,783	777,502	1,557,757
J1-14-20-3	1,979	19	396	428	47,902	444	348,312	348,769	699,491
J1-14-20-4	2,631	23	1,080	1,116	153,648	654	634,924	635,591	1,273,787
J2-10-35-0	4,486	10	4,144	4,163	1,934,260	1,404	3,505,514	3,506,927	7,018,322
J2-10-35-1	4,572	12	3,818	3,839	1,722,235	1,416	3,569,796	3,571,221	7,146,996
J2-10-35-2	4,892	12	3,752	3,774	1,635,389	1,560	3,811,514	3,812,784	7,630,440
J2-10-35-3	4,328	12	2,823	2,844	1,112,524	1,798	2,875,373	2,877,090	5,758,490
J2-10-35-4	3,992	12	3,343	3,365	1,284,670	1,380	2,681,176	2,682,566	5,369,104
J2-12-35-0	6,240	11	5,387	5,409	3,007,118	1,704	6,493,882	6,495,597	12,997,412
J2-12-35-1	6,100	13	4,927	4,951	2,308,719	1,752	5,234,224	5,235,987	10,478,052
J2-12-35-2	5,852	16	5,148	5,174	2,207,902	2,376	4,829,349	4,831,735	9,669,302
J2-12-35-3	5,420	15	4,233	4,259	1,791,202	1,836	4,150,239	4,152,086	8,309,570
J2-12-35-4	6,256	14	3,747	3,772	1,971,096	1,480	5,439,275	5,440,766	10,887,766
J2-14-35-0	8,718	14	7,049	7,076	4,184,740	2,736	10,161,001	10,163,750	20,336,192
J2-14-35-1	8,620	15	6,318	6,346	3,606,054	2,768	9,309,051	9,311,832	18,632,258
J2-14-35-2	7,484	17	6,027	6,055	2,871,151	2,940	6,818,098	6,821,049	13,649,560
J2-14-35-3	7,552	15	5,779	5,807	2,994,843	2,492	7,561,900	7,564,405	15,136,336
J2-14-35-4	8,056	12	5,285	5,310	2,900,147	2,422	8,157,073	8,159,508	16,327,046
poly1a	5,620	34	4,432	4,481	1,366,645	2,072	3,285,681	3,287,768	6,581,126
poly1b	6,858	24	5,554	5,593	2,002,020	2,515	4,922,991	4,925,521	9,857,870
poly1c	4,437	20	2,879	2,914	557,763	534	1,513,472	1,514,021	3,032,449
poly1d	4,527	32	2,963	3,010	665,693	1,530	1,947,175	1,948,720	3,901,937
poly1e	3,638	20	2,625	2,660	545,425	1,062	1,301,446	1,302,523	2,608,654
jakobs1	7,460	44	1,790	1,856	415,306	1,628	2,846,369	2,848,019	5,703,454

Table 4 – Comparison between clique covering and dotted-board models (time limit of 18,000 s).

Instance	Clique covering model					Dotted-board model				
	UB	LB	Gap (%)	Time (s)	Nodes	UB	LB	Gap (%)	Time (s)	Nodes
three	6	6.00	0.00	0.01	0	6	6.00	0.00	0.01	0
threep2	10	10.00	0.00	0.01	0	10	10.00	0.00	0.01	0
threep2w9	8	8.00	0.00	0.01	0	8	8.00	0.00	0.02	0
threep3	14	14.00	0.00	0.01	0	14	14.00	0.00	0.02	0
threep3w9	12	12.00	0.00	0.05	0	12	12.00	0.00	0.10	13
shapes4	24	24.00	0.00	0.05	0	24	24.00	0.00	0.87	0
shapes8	26	26.00	0.00	9.07	0	26	26.00	0.00	94.33	539
SHAPES2	14	14.00	0.00	1.09	0	14	14.00	0.00	4.73	0
SHAPES4	25	25.00	0.00	10,869.40	37,344	25	24.00	4.00	TL	31,134
SHAPES5	30	28.00	6.67	TL	36,299	30	26.00	13.33	TL	45,834
SHAPES7	44	35.00	20.45	TL	6,970	46	34.00	26.09	TL	3
SHAPES9	52	38.00	26.92	TL	1,376	54	38.00	29.63	TL	0
SHAPES15	67	47.00	29.85	TL	0	67	47.00	29.85	TL	0
blasz2	26	26.00	0.00	18.47	3,475	26	26.00	0.00	51.79	2,449
BLAZEWCZ1	8	8.00	0.00	0.00	0	8	8.00	0.00	0.00	0
BLAZEWCZ2	14	14.00	0.00	4.19	0	14	14.00	0.00	17.98	0
BLAZEWCZ3	20	20.00	0.00	1,144.03	16,250	20	20.00	0.00	655.42	4,424
BLAZEWCZ4	27	27.00	0.00	11,936.00	107,195	27	26.00	3.70	TL	55,979
BLAZEWCZ5	34	32.00	5.88	TL	75,181	34	32.00	5.88	TL	46,038
RCO1	8	8.00	0.00	0.00	0	8	8.00	0.00	0.00	0
RCO2	15	15.00	0.00	1.33	0	15	15.00	0.00	2.00	0
RCO3	22	22.00	0.00	140.11	5,754	22	22.00	0.00	260.64	3,394
RCO4	29	28.00	3.45	TL	785,086	29	28.00	3.45	TL	407,168
RCO5	36	35.00	2.78	TL	388,383	36	35.00	2.78	TL	139,837
artif1_2	7	7.00	0.00	0.31	0	7	7.00	0.00	1.36	0
artif2_4	13	13.00	0.00	1,327.72	12,783	13	13.00	0.00	1,795.71	4,511
artif3_6	19	18.00	5.26	TL	61,600	19	17.00	10.53	TL	12,462
artif4_8	25	23.00	8.00	TL	22,680	27	22.00	18.52	TL	14
artif5_10	32	28.00	12.50	TL	4,363	-	27.00	-	TL	28
artif6_12	39	33.00	15.38	TL	2,032	41	33.00	19.51	TL	7
artif7_14	45	38.00	15.56	TL	470	-	38.00	-	TL	0
artif	51	42.00	17.65	TL	0	-	41.00	-	TL	0
shirts1_2	13	13.00	0.00	0.00	0	13	13.00	0.00	0.01	0
shirts2_4	17	17.00	0.00	50.59	46	17	17.00	0.00	167.37	128
shirts3_6	24	24.00	0.00	399.71	108	25	24.00	4.00	TL	21,519
shirts4_8	33	31.00	6.06	TL	6,552	35	31.00	11.43	ML	1,718
shirts5_10	41	37.09	9.53	TL	3,787	-	37.00	-	ML	0
dagli1	23	23.00	0.00	111.53	0	23	23.00	0.00	294.15	0
fu5	18	18.00	0.00	0.14	0	18	18.00	0.00	1.04	0
fu6	23	23.00	0.00	2.40	0	23	23.00	0.00	12.88	0
fu7	24	24.00	0.00	1.27	0	24	24.00	0.00	17.23	0
fu8	24	24.00	0.00	7.08	0	24	24.00	0.00	34.75	0
fu9	25	25.00	0.00	126.13	223	27	24.00	11.11	ML	6,229
fu10	29	29.00	0.00	984.43	235	-	28.00	-	ML	6
fu	34	31.08	8.59	TL	6,637	-	-	-	ML	0
J1-10-10-0	18	18.00	0.00	0.79	0	18	18.00	0.00	2.99	0
J1-10-10-1	17	17.00	0.00	1.02	0	17	17.00	0.00	14.83	11
J1-10-10-2	20	20.00	0.00	0.84	0	20	20.00	0.00	9.51	0
J1-10-10-3	21	21.00	0.00	13.93	177	21	21.00	0.00	124.67	725
J1-10-10-4	13	13.00	0.00	2.34	225	13	13.00	0.00	38.45	181
J1-12-20-0	12	12.00	0.00	1.12	0	12	12.00	0.00	49.27	0
J1-12-20-1	10	10.00	0.00	1.51	0	10	10.00	0.00	32.56	71
J1-12-20-2	12	12.00	0.00	3.25	0	12	12.00	0.00	90.19	0
J1-12-20-3	8	8.00	0.00	0.72	0	8	8.00	0.00	22.94	0
J1-12-20-4	13	13.00	0.00	9.20	59	13	13.00	0.00	409.98	33
J1-14-20-0	12	12.00	0.00	31.62	4,424	12	12.00	0.00	235.25	583
J1-14-20-1	12	12.00	0.00	60.23	2,825	12	12.00	0.00	384.44	3,401
J1-14-20-2	14	14.00	0.00	71.87	2,008	14	14.00	0.00	9,045.74	7,967
J1-14-20-3	10	10.00	0.00	1.91	0	10	10.00	0.00	212.28	454
J1-14-20-4	14	14.00	0.00	22.62	617	14	14.00	0.00	467.39	473
J2-10-35-0	24	24.00	0.00	2,530.36	4,491	26	22.00	15.38	ML	655
J2-10-35-1	25	22.00	12.00	TL	29,135	25	22.00	12.00	TL	30,685
J2-10-35-2	23	23.00	0.00	6,137.75	5,165	24	21.00	12.50	ML	4,165
J2-10-35-3	21	21.00	0.00	1,257.77	5,545	22	18.00	18.18	ML	58,216
J2-10-35-4	19	18.00	5.26	TL	18,138	20	17.00	15.00	ML	1,356
J2-12-35-0	30	25.01	16.63	TL	4,200	-	25.00	-	ML	0
J2-12-35-1	25	23.00	8.00	TL	3,309	28	23.00	17.86	TL	5
J2-12-35-2	24	21.00	12.50	TL	13,217	28	21.00	25.00	ML	16
J2-12-35-3	21	19.00	9.52	TL	14,885	23	19.00	17.39	ML	53
J2-12-35-4	26	24.00	7.69	TL	5,831	29	24.00	17.24	TL	7
J2-14-35-0	32	26.00	18.75	TL	2	-	26.00	-	ML	0
J2-14-35-1	32	25.00	21.88	TL	7	-	25.00	-	ML	0
J2-14-35-2	28	23.00	17.86	TL	1,385	30	23.00	23.33	TL	7
J2-14-35-3	27	22.00	18.52	TL	4,963	-	22.00	-	ML	0
J2-14-35-4	28	24.00	14.29	TL	3,460	-	24.00	-	TL	0
poly1a	16	14.00	12.50	TL	35,978	18	14.00	22.22	TL	0
poly1b	19	16.00	15.79	TL	12,335	20	16.00	20.00	ML	24
poly1c	13	13.00	0.00	213.11	0	13	13.00	0.00	153.30	0
poly1d	13	11.00	15.38	TL	112,747	13	11.00	15.38	TL	35,261
poly1e	12	11.00	8.33	TL	26,271	12	11.00	8.33	TL	66,842
jakobs1	11	11.00	0.00	165.87	2,079	12	11.00	8.33	TL	11,947
#Feasible	81					70				
#Optimal	48					40				

Table 5 – Comparison between clique covering and dotted-board models (time limit of 3,600 s).

Instance	Clique covering model					Dotted-board model				
	UB	LB	Gap (%)	Time (s)	Nodes	UB	LB	Gap (%)	Time (s)	Nodes
three	6	6.00	0.00	0.01	0	6	6.00	0.00	0.01	0
threep2	10	10.00	0.00	0.01	0	10	10.00	0.00	0.01	0
threep2w9	8	8.00	0.00	0.01	0	8	8.00	0.00	0.02	0
threep3	14	14.00	0.00	0.01	0	14	14.00	0.00	0.02	0
threep3w9	12	12.00	0.00	0.06	0	12	12.00	0.00	0.10	13
shapes4	24	24.00	0.00	0.05	0	24	24.00	0.00	0.88	0
shapes8	26	26.00	0.00	8.92	0	26	26.00	0.00	123.97	539
SHAPES2	14	14.00	0.00	1.09	0	14	14.00	0.00	4.69	0
SHAPES4	25	24.00	4.00	TL	11,873	26	22.00	15.38	TL	9,532
SHAPES5	31	27.01	12.86	TL	3,954	32	26.00	18.75	TL	2,049
SHAPES7	45	34.00	24.44	TL	12	-	34.00	-	TL	3
SHAPES9	54	38.00	29.63	TL	0	-	38.00	-	TL	0
SHAPES15	-	47.00	-	TL	0	-	47.00	-	TL	0
blasz2	26	26.00	0.00	19.62	3,475	26	26.00	0.00	51.18	2,449
BLAZEWCZ1	8	8.00	0.00	0.00	0	8	8.00	0.00	0.00	0
BLAZEWCZ2	14	14.00	0.00	4.17	0	14	14.00	0.00	16.98	0
BLAZEWCZ3	20	20.00	0.00	1,139.96	16,250	20	20.00	0.00	617.17	4,424
BLAZEWCZ4	27	26.00	3.70	TL	54,490	28	26.00	7.14	TL	13,132
BLAZEWCZ5	34	32.00	5.88	TL	32,324	35	32.00	8.57	TL	3,992
RCO1	8	8.00	0.00	0.00	0	8	8.00	0.00	0.00	0
RCO2	15	15.00	0.00	1.07	0	15	15.00	0.00	1.99	0
RCO3	22	22.00	0.00	141.86	5,754	22	22.00	0.00	261.42	3,394
RCO4	29	28.00	3.45	TL	208,796	29	28.00	3.45	TL	81,555
RCO5	36	35.00	2.78	TL	80,235	36	35.00	2.78	TL	25,256
artif1_2	7	7.00	0.00	0.28	0	7	7.00	0.00	1.55	0
artif2_4	13	13.00	0.00	1,340.63	12,783	13	13.00	0.00	1,870.03	4,511
artif3_6	19	17.00	10.53	TL	11,097	20	17.00	15.00	TL	1,164
artif4_8	26	22.11	14.98	TL	3,780	28	22.00	21.43	TL	0
artif5_10	34	28.00	17.65	TL	17	-	27.00	-	TL	28
artif6_12	40	33.00	17.50	TL	0	-	33.00	-	TL	7
artif7_14	-	38.00	-	TL	0	-	38.00	-	TL	0
artif	-	41.00	-	TL	0	-	41.00	-	TL	0
shirts1_2	13	13.00	0.00	0.02	0	13	13.00	0.00	0.01	0
shirts2_4	17	17.00	0.00	47.77	46	17	17.00	0.00	253.67	86
shirts3_6	24	24.00	0.00	497.68	108	26	24.00	7.69	TL	3,869
shirts4_8	33	31.00	6.06	TL	455	35	31.00	11.43	TL	0
shirts5_10	-	37.00	-	TL	0	-	37.00	-	TL	0
dagli1	23	23.00	0.00	100.73	0	23	23.00	0.00	293.41	0
fu5	18	18.00	0.00	0.11	0	18	18.00	0.00	1.05	0
fu6	23	23.00	0.00	2.11	0	23	23.00	0.00	13.02	0
fu7	24	24.00	0.00	1.30	0	24	24.00	0.00	17.50	0
fu8	24	24.00	0.00	6.90	0	24	24.00	0.00	35.80	0
fu9	25	25.00	0.00	129.74	223	27	24.00	11.11	ML	5,678
fu10	29	29.00	0.00	1,032.96	235	-	28.00	-	ML	0
fu	37	31.00	16.22	TL	0	-	-	-	ML	0
J1-10-10-0	18	18.00	0.00	0.63	0	18	18.00	0.00	3.43	0
J1-10-10-1	17	17.00	0.00	1.04	0	17	17.00	0.00	16.00	11
J1-10-10-2	20	20.00	0.00	0.86	0	20	20.00	0.00	7.34	0
J1-10-10-3	21	21.00	0.00	13.99	177	21	21.00	0.00	125.31	725
J1-10-10-4	13	13.00	0.00	2.37	225	13	13.00	0.00	39.01	181
J1-12-20-0	12	12.00	0.00	1.13	0	12	12.00	0.00	48.57	0
J1-12-20-1	10	10.00	0.00	1.53	0	10	10.00	0.00	33.37	71
J1-12-20-2	12	12.00	0.00	3.28	0	12	12.00	0.00	91.20	0
J1-12-20-3	8	8.00	0.00	0.73	0	8	8.00	0.00	22.76	0
J1-12-20-4	13	13.00	0.00	9.31	59	13	13.00	0.00	328.09	92
J1-14-20-0	12	12.00	0.00	32.09	4,424	12	12.00	0.00	160.92	0
J1-14-20-1	12	12.00	0.00	59.86	2,825	12	12.00	0.00	390.02	3,401
J1-14-20-2	14	14.00	0.00	71.53	2,008	15	13.00	13.33	TL	9,280
J1-14-20-3	10	10.00	0.00	1.95	0	10	10.00	0.00	233.32	454
J1-14-20-4	14	14.00	0.00	22.07	617	14	14.00	0.00	1,773.62	3,714
J2-10-35-0	25	22.06	11.78	TL	2,352	27	22.00	18.52	TL	0
J2-10-35-1	25	22.00	12.00	TL	3,945	26	22.00	15.38	TL	195
J2-10-35-2	23	21.05	8.48	TL	4,085	25	21.00	16.00	TL	3,354
J2-10-35-3	21	21.00	0.00	1,167.54	5,545	22	18.00	18.18	TL	5,135
J2-10-35-4	20	18.00	10.00	TL	4,179	20	17.00	15.00	ML	1,061
J2-12-35-0	31	25.00	19.35	TL	2	-	25.00	-	ML	0
J2-12-35-1	27	23.00	14.81	TL	209	-	23.00	-	TL	5
J2-12-35-2	26	21.00	19.23	TL	5	28	21.00	25.00	TL	4
J2-12-35-3	24	19.00	20.83	TL	1,583	25	19.00	24.00	TL	8
J2-12-35-4	27	24.00	11.11	TL	984	29	24.00	17.24	TL	2
J2-14-35-0	-	26.00	-	TL	0	-	-	-	ML	0
J2-14-35-1	33	25.00	24.24	TL	0	-	25.00	-	TL	0
J2-14-35-2	33	23.00	30.30	TL	0	-	23.00	-	ML	0
J2-14-35-3	28	22.00	21.43	TL	0	-	22.00	-	ML	0
J2-14-35-4	29	24.00	17.24	TL	0	-	24.00	-	TL	0
poly1a	17	14.00	17.65	TL	2,320	18	14.00	22.22	TL	0
poly1b	20	16.00	20.00	TL	0	21	16.00	23.81	TL	0
poly1c	13	13.00	0.00	152.25	0	13	13.00	0.00	223.20	0
poly1d	13	11.00	15.38	TL	9,010	-	11.00	-	TL	2
poly1e	12	11.00	8.33	TL	5,867	13	11.00	15.38	TL	6,678
jakobs1	11	11.00	0.00	181.30	2,079	-	11.00	-	TL	0
#Feasible	76					62				
#Optimal	45					39				

execution time (in seconds) and number of explored nodes in the search tree. Flags TL and ML indicate, respectively, the execution time or memory limit has been reached. Symbol - indicates no feasible solution has been found. The bottom rows show the number of feasible and optimal solutions found for each model.

The overall results show the superiority of the clique covering model in terms of feasible and optimal solutions (upper bounds), lower bounds, optimality gap, runtime and number of nodes explored. Furthermore, the model could tackle larger instances than the dotted-board model, because it drastically reduced the number of constraints, hence, the required memory of the solver. It also found an optimal solution for instance jakobs1, subject to integer discretization. This is the first time an instance with more than 20 piece types has been solved to optimality. However, the dotted-board model found an optimal solution for some instances in less time (BLAZEWCZ3, poly1c) or exploring fewer nodes (SHAPES4, blasz2, BLAZEWCZ3, RCO3, artif2_4, J1-10-10-4, J1-12-20-4, J1-14-20-0, J1-14-20-4) than the clique covering model (see Table 4).

Although the clique covering model can solve larger instances than the dotted-board model, it could not find a feasible solution for five instances (SHAPES15, artif7_14, artif, shirts5_10 and J2-14-35-0) within a time limit of 3,600 seconds (see Table 5). Nevertheless, the proposed model found a feasible solution for all instances within a time limit of 18,000 seconds (see Table 4).

The best solutions obtained by the clique covering model are shown in Appendix A.

2.5 Conclusions and further work

We have proposed a clique covering MIP model for the irregular strip packing problem based on the dotted-board model (TOLEDO *et al.*, 2013). It can solve instances with up to 25 pieces of 22 piece types to optimality, subject to the board discretization. Moreover, it obtained better performance than the dotted-board model for most instances and solved larger instances to optimality in comparison to state-of-the-art models. We have also introduced valid inequalities for the clique covering model and developed an iterative method to calculate a better lower bound and enhance the formulation. For further work, we suggest adding classicalmatemática valid inequalities from the stable set polytope (REBENNACK; REINELT; PARDALOS, 2012) to improve the linear relaxation of the clique covering model. Such inequalities, as well the ones proposed in this paper can be used in the development of a branch-and-cut algorithm for the problem.

MIP MODELS FOR THE IRREGULAR STRIP PACKING PROBLEM: NEW SYMMETRY BREAKING CONSTRAINTS

This chapter is based on the short paper available online at APMOD 2016 conference proceeding website: <<https://doi.org/10.1051/itmconf/20171400005>>. RODRIGUES, M. O.; CHERRI, L. H.; MUNDIM, L. R. *MIP models for the irregular strip packing problem: new symmetry breaking constraints. ITM Web Conf.*, 14. p. 5. [Rodrigues, Cherri and Mundim \(2017\)](#).

3.1 Introduction

The irregular strip packing problem (also known as irregular open dimensional problem) consists in cutting irregular pieces from a rectangular board with fixed width and infinite length. In the problem, pieces cannot overlap and each piece must be inside the board. The objective is to minimize the board length used to cut the pieces. The problem is both economical and environmental relevant, since reducing material consumption decreases production costs and minimizes wastes.

Applications and its difficult Such problems appear in several several industries, e.g. , shoes manufacturing, metalworking,

The irregular strip packing problem is NP-hard ([NIELSEN; ODGAARD, 2003](#)) and the methods proposed to solve it are mainly heuristics. The best solutions for large benchmark instances were found in ([ELKERAN, 2013](#)). He proposed a guided cuckoo search algorithm to solve the problem. In his method, the best fit between two pieces is found using a pairwise clustering and the n the guided cuckoo search algorithm is applied to find complete solutions.

Some mixed integer programming (MIP) models were proposed to represent the problem. [Fischetti and Luzzi \(2009\)](#) were the first to propose a MIP model to the problem. [Alvarez-Valdes,](#)

Martinez and Tamarit (2013) defined some structures used in (FISCHETTI; LUZZI, 2009) and developed a branch-and-bound algorithm to solve the model. Santoro and Lemos (2015) proposed a MIP model to convex polygons with some limitations: the polygons can have up to eight edges and their opposite edges must be parallel. Cherri *et al.* (2016) proposed two models that allow rotation of pieces. The models differ in the geometric structures used to avoid overlap among pieces. Toledo *et al.* (2013) proposed the dotted-board model where the pieces were placed in a finite set of dots inside the board. The optimality of the model is subject to the discretization used to generate the dots. In order to combine the accuracy of continuous models with the advantages of the dotted-board model, Leao *et al.* (2016) proposed a semi-continuous model where the pieces can be placed in continuous positions in x-axis and in discrete positions in y-axis.

Here, new symmetry breaking constraints for the MIP model described in (CHERRI *et al.*, 2016) are proposed. Computational results show the performance of the model is improved in various aspects such as lower bounds, number of nodes explored and run-time.

3.2 Cherri et al. (2016) model

3.3 MIP model

In this section, we define the irregular strip packing problem and review the No-Fit Polygon Covering Model (NFP-CM) proposed in (CHERRI *et al.*, 2016).

Given a set of N pieces and a board with fixed width W , the irregular strip packing problem consists in minimizing the length z of the board used to cut all pieces. Each piece is defined by a set of vertices in clockwise orientation and one of the vertices is chosen to be its reference point. Let $w_i^{\min}$ ($w_i^{\max}$) be the vertical distance between the reference point of piece i and the vertex of the piece that is nearest (farthest) to the origin. Likewise, let $\ell_i^{\min}$ ($\ell_i^{\max}$) be the distance of the reference point of piece i and the leftmost (rightmost) vertex of the piece. Consider (x_i, y_i) the variables that define the translation of the reference point of piece i to place it in the board. Figure 12 illustrates the distance parameters of a piece in NFP-CM.

Figure 12 – Representation of a piece and its distance parameters in NFP-CM.

Source: Elaborated by the author.

The main difficulty to solve the problem is the non-overlap constraints. In NFP-CM the

no-fit polygon is used to prevent overlap. For a review about non-overlap techniques and the geometry of the problem, see (BENNELL; OLIVEIRA, 2008).

Cherri *et al.* (2016) defined the following binary variables to prevent overlap:

$$\xi_{ij}^k = \begin{cases} 1, & \text{if piece } j \text{ is on the right side of edge } k \text{ of } NFP_{ij}; \\ 0, & \text{otherwise.} \end{cases} \quad (3.1)$$

where NFP_{ij} is the no-fit polygon between pieces i and j . If piece i is on the right side of at least one edge $k = 1, \dots, K_{ij}$ of NFP_{ij} , there is no overlap between pieces i and j .

NFP-CM is given by:

$$\min \quad z \quad (3.2)$$

s.t.

$$\ell_i^{\min} \leq x_i \leq z - \ell_i^{\max}, \quad i = 1, \dots, N, \quad (3.3)$$

$$w_i^{\min} \leq y_i \leq W - w_i^{\max}, \quad i = 1, \dots, N, \quad (3.4)$$

$$a_{ij}^k(x_i - x_j) + b_{ij}^k(y_i - y_j) \leq c_{ij}^k + M_{ij}^k(1 - \xi_{ij}^k), \quad 1 \leq i < j \leq N, k = 1, \dots, K_{ij}, \quad (3.5)$$

$$\sum_{k=1}^{K_{ij}} \xi_{ij}^k = 1, \quad 1 \leq i < j \leq N, \quad (3.6)$$

$$\xi_{ij}^k \in \{0, 1\}, \quad 1 \leq i < j \leq N, k = 1, \dots, K_{ij}. \quad (3.7)$$

Objective function (3.2) minimizes the length used to cut all pieces. Constraints (3.3) and (3.4) impose the pieces are placed inside the board. Constraints (3.5) prevent overlap among pieces. Constraints (3.6) ensure each piece is on the right side of an edge of the no-fit polygon while constraints (3.7) define the domain of binary variables.

3.4 Symmetry breaking constraints

NFP-CM (3.2)-(3.7) may have infinite symmetric solutions. Therefore, identical or equivalent solutions can be found several times by the enumeration scheme. In this section symmetric breaking constraints are proposed. The constraints can improve the linear relaxation of the model and reduce the number of identical solutions.

3.4.1 Partitioning the solution space

The main drawback of NFP-CM is non-overlap constraints (3.5). The definition of variables ξ_{ij}^k generates overlap between various regions, as shown in Figure 13(a). Dotted regions are common to two binary variables and hatched regions are common to three variables.

Such redundancies on ξ_{ij}^k values can be avoided by adding constraints (3.10) to the model. The constraints ensure the reference point of piece j is on the left side of the line defined by edge

k' , where edge k' precedes edge k in the no-fit polygon. Therefore, the reference point of piece j can activate only one ξ_{ij}^k variable. Figure 13(b) illustrates the partitioning obtained from a no-fit polygon.

$$\bar{a}_{ij}^{k'}(x_i - x_j) + \bar{b}_{ij}^{k'}(y_i - y_j) \leq \bar{c}_{ij}^{k'} + \bar{M}_{ij}^{k'}(1 - \xi_{ij}^k), \quad 1 \leq i < j \leq N, k = 1, \dots, K_{ij}, \quad (3.8)$$

3.4.2 Partitioning the solution space

The main drawback of NFP-CM is non-overlap constraints (3.5). Such redundancies on ξ_{ij}^k values can be avoided by adding constraints (3.10) to the model. The constraints ensure the reference point of piece j is on the left side of the line defined by edge k' , where edge k' precedes edge k in the no-fit polygon. Therefore, the reference point of piece j can activate only one ξ_{ij}^k variable. Figure 13(b) illustrates the partitioning obtained from a no-fit polygon.

$$\bar{a}_{ij}^{k'}(x_i - x_j) + \bar{b}_{ij}^{k'}(y_i - y_j) \leq \bar{c}_{ij}^{k'} + \bar{M}_{ij}^{k'}(1 - \xi_{ij}^k), \quad 1 \leq i < j \leq N, k = 1, \dots, K_{ij}, \quad (3.9)$$

The definition of variables ξ_{ij}^k generates overlap between various regions, as shown in Figure 13(a). Dotted regions are common to two binary variables and hatched regions are common to three variables.

Figure 13 – Non-overlap regions. (a) NFP-CM model. (b) Proposed partitioning.

Source: Elaborated by the author.

Such redundancies on ξ_{ij}^k values can be avoided by adding constraints (3.10) to the model. The constraints ensure the reference point of piece j is on the left side of the line defined by edge k' , where edge k' precedes edge k in the no-fit polygon. Therefore, the reference point of piece j can activate only one ξ_{ij}^k variable. Figure 13(b) illustrates the partitioning obtained from a no-fit polygon.

$$\bar{a}_{ij}^{k'}(x_i - x_j) + \bar{b}_{ij}^{k'}(y_i - y_j) \leq \bar{c}_{ij}^{k'} + \bar{M}_{ij}^{k'}(1 - \xi_{ij}^k), \quad 1 \leq i < j \leq N, k = 1, \dots, K_{ij}, \quad (3.10)$$

3.4.3 Connecting pieces

In a feasible solution for the irregular strip packing problem, some pieces may not touch each other. However, one can translate the pieces to connect them, obtaining another solution

with the same quality. Consider the following variables:

$$\phi_{ij} = \begin{cases} 1, & \text{if piece } i \text{ is connected to piece } j; \\ 0, & \text{otherwise.} \end{cases} \quad (3.11)$$

Constraints (3.12) combined with (3.5) ensure if ϕ_{ij} is equal to 1, the pieces i and j are connected.

$$a_{ij}^k(x_i - x_j) + b_{ij}^k(y_i - y_j) \geq c_{ij}^k - \underline{M}_{ij}^k(1 - \phi_{ij}), \quad 1 \leq i < j \leq N, k = 1, \dots, K_{ij}. \quad (3.12)$$

To ensure each piece is connected with another piece, the following constraints are imposed:

$$\sum_{j=1}^{i-1} \phi_{ji} + \sum_{j=i+1}^N \phi_{ij} \geq 1, \quad i = 1, \dots, N. \quad (3.13)$$

Note that constraints (3.13) do not ensure all pieces are connected as a single component. For example, it may have two groups of connected pieces with no interconnection. To enforce all pieces are connected, one can impose all cut sets of pieces are connected, as defined by constraints (3.14).

$$\sum_{(i,j) \in \delta(S) | i < j} \phi_{ij} \geq 1, \quad \forall S \subset N, S \neq \emptyset \quad (3.14)$$

where $\delta(S)$ is the cutset of S .

Although constraints (3.14) impose connectivity, they introduce an exponential number of constraints in the model and therefore hamper its resolution.

3.4.4 Connecting pieces with the board

In the optimal solution, at least one piece can touch the bottommost position of the board. In fact, if there is a solution that does not satisfy this statement, one can translate a piece down until it touches the board, obtaining a solution with the same quality.

Let ω_i be a binary variable, defined as follows:

$$\omega_i = \begin{cases} 1, & \text{if piece } i \text{ is in the bottommost position of the board;} \\ 0, & \text{otherwise.} \end{cases}$$

Constraints (3.15) and (3.16) ensure at least one piece is placed in the bottommost position in the board.

$$y_i \leq w_i^{\min} + (W - w_i^{\max} - w_i^{\min})(1 - \omega_i), \quad i = 1, \dots, N, \quad (3.15)$$

$$\sum_{i=1}^N \omega_i = 1. \quad (3.16)$$

3.4.5 Pieces with same shape

Symmetric solutions can be found if there are multiple pieces with same type (shape). For example, if there are two pieces with same type, all the solutions can be duplicated in an enumeration scheme. To avoid such problem, the pieces of same type can be ordered. Consider T_i the type of piece i , to eliminate the symmetry on the placement of pieces of same type, the following constraints are imposed.

$$y_i \leq y_j, \quad i = 1, \dots, N, j = \min\{k = i + 1, \dots, N | T_k = T_i\}. \quad (3.17)$$

Note that constraints (3.17), except in the case where the pieces are placed in the same y -position, eliminate symmetry of identical pieces.

Some values of ω_i can be inferred for pieces with same type using constraints (3.15)-(3.17). Specifically, for a piece with several copies, only the smallest index constraint (3.15) can be active. Constraints (3.18) ensure only the smallest index copy of each piece type can activate constraints (3.15).

$$\omega_j = 0, \quad i = 1, \dots, N, j = \min\{k = i + 1, \dots, N | T_k = T_i\}. \quad (3.18)$$

3.5 Results

In this section we present computational experiments to evaluate the performance of the proposed symmetry breaking constraints. The experiments were run on a computer with Intel® Core™ i7-2600 @ 3.40GHz CPU and 16 GB RAM under Ubuntu 14.04 LTS 64 bits operating system. The instances were solved by IBM ILOG CPLEX 12.6 solver using default parameters and a time limit of 3,600 seconds.

Table 6 presents the main characteristics of the evaluated instances: the name, the author(s) who proposed it, the total number of pieces, the number of piece types, the average number of vertices, and the upper bound (UB). The total number of pieces ranges between 3 and 17, with an average of 8.6 pieces, and an average number of 4.02 vertices per instance. Additionally, we present the upper bound (UB) used to estimate the big-M coefficients in constraints (3.5), for more details see (CHERRI *et al.*, 2016). UB was calculated using the bottom-left heuristic proposed in (GOMES; OLIVEIRA, 2006) with one thousand random input sequences.

Table 7 reports the computational experiments of NFP-CM proposed in (CHERRI *et al.*, 2016) and the improved NFP-CM, i.e., NFP-CM with symmetry breaking constraints (3.10) and (3.15)-(3.18). Improved NFP-CM obtained better or equal lower bounds (LB) for all instances. Moreover, improved NFP-CM proved optimality for nineteen instances, four more instances than NFP-CM. In average, improved NFP-CM was 35.70% faster than NFP-CM to prove optimality.

Table 6 – Features of benchmark instances.

Instance	Author(s)	Pieces	Piece types	Avg. vertices	UB
jones1	Jones (2014)	4	4	3.25	5.00
jones2	Jones (2014)	6	6	3.83	10.00
three	Alvarez-Valdes, Martinez and Tamarit (2013)	3	3	3.67	6.67
threep2	Alvarez-Valdes, Martinez and Tamarit (2013)	6	3	3.67	9.67
threep2w9	Alvarez-Valdes, Martinez and Tamarit (2013)	6	3	3.67	8.67
threep3	Alvarez-Valdes, Martinez and Tamarit (2013)	9	3	3.67	14.00
threep3w9	Alvarez-Valdes, Martinez and Tamarit (2013)	9	3	3.67	12.00
fu5	Alvarez-Valdes, Martinez and Tamarit (2013)	5	4	3.20	37.00
fu6	Alvarez-Valdes, Martinez and Tamarit (2013)	6	5	3.33	17.89
fu7	Alvarez-Valdes, Martinez and Tamarit (2013)	7	6	3.43	24.00
fu8	Alvarez-Valdes, Martinez and Tamarit (2013)	8	7	3.50	24.00
fu9	Alvarez-Valdes, Martinez and Tamarit (2013)	9	8	3.44	24.00
fu10	Alvarez-Valdes, Martinez and Tamarit (2013)	10	9	3.50	28.00
fu	Alvarez-Valdes, Martinez and Tamarit (2013)	12	11	3.42	30.00
rco1	Ribeiro, Carravilla and Oliveira (1999)	7	7	5.14	8.00
rco2	Ribeiro, Carravilla and Oliveira (1999)	14	7	5.14	15.00
santoro1	Santoro and Lemos (2015)	11	11	8.00	28.00
santoro2	Santoro and Lemos (2015)	11	10	4.55	6.20
santoro3	Santoro and Lemos (2015)	12	8	4.33	6.50
santoro4	Santoro and Lemos (2015)	17	11	4.00	19.00

However, improved NFP-CM was slower for instance santoro4. The experiments were run on a computer with Intel® Core™ i7-2600 @ 3.40GHz CPU and 16 GB RAM under Ubuntu 14.04 LTS 64 bits operating system. The instances were solved by IBM ILOG CPLEX 12.6 solver using default parameters and a time limit of 3,600 seconds.

Considering nodes explored by CPLEX, three instances were solved at the root node. For the remaining instances, improved NFP-CM explored fewer nodes for fourteen cases. The biggest reduction of nodes explored to prove optimality was obtained for instance santoro1, where NFP-CM explored 185.36 times the amount of nodes of improved NFP-CM. Furthermore, improved NFP-CM solved all instances santoro faster than the original paper ([SANTORO; LEMOS, 2015](#)). We also highlight improved NFP-CM is the first MIP model to prove optimality for instances threep3, threep3w9 and fu in less than 1,200 seconds.

3.6 Conclusions and further work

In this paper we have proposed symmetry breaking constraints for the irregular strip packing problem. Although this new set of constraints was added to NFP-CM proposed in ([CHERRI *et al.*, 2016](#)), the ideas presented here can be applied to other models.

Computational experiments show the relevance of this new set of constraints. In general, the symmetry breaking constraints improve lower bounds and reduce run-time and the number of nodes explored to prove optimality.

For further work, we intend to introduce symmetry breaking constraints for instances with non-convex pieces and study branching priorities for binary variables.

Table 7 – Comparison between MIP models.

Instance	NFP-CM				Improved NFP-CM			
	LB	UB	Time (s)	Nodes	LB	UB	Time (s)	Nodes
jones1	5.00	5.00	0	0	5.00	5.00	0	0
jones2	10.00	10.00	0	0	10.00	10.00	0	0
three	6.00	6.00	0	0	6.00	6.00	0	0
threep2	9.33	9.33	2	21,360	9.33	9.33	1	12,963
threep2w9	8.00	8.00	6	107,113	8.00	8.00	1	15,217
threep3	10.06	13.53	3,600	24,408,906	13.53	13.53	928	9,711,358
threep3w9	8.75	11.00	3,600	33,442,609	11.00	11.00	1,113	9,036,051
fu5	17.89	17.89	0	336	17.89	17.89	0	235
fu6	23.00	23.00	0	428	23.00	23.00	0	452
fu7	24.00	24.00	0	1,324	24.00	24.00	0	1,481
fu8	24.00	24.00	0	851	24.00	24.00	0	3,127
fu9	25.00	25.00	7	83,709	25.00	25.00	4	32,911
fu10	28.69	28.69	301	2,653,576	28.69	28.69	33	253,314
fu	28.00	33.14	3,600	18,761,708	33.14	33.14	511	2,539,903
rco1	8.00	8.00	1	14,127	8.00	8.00	1	12,008
rco2	6.67	-	3,600	17,116,136	9.00	-	3,600	10,534,103
santoro1	26.85	28.00	3,600	49,058,509	28.00	28.00	41	264,666
santoro2	6.00	6.00	474	6,958,452	6.00	6.00	19	120,888
santoro3	6.00	6.00	5	30,938	6.00	6.00	5	17,955
santoro4	15.00	15.00	24	178,258	15.00	15.00	52	131,688
Optimal	15			19				

INTEGER PROGRAMMING MODELS FOR THE QUASI-POLYOMINO STRIP PACKING PROBLEM

4.1 Introduction

[Golomb \(1954\)](#) defined the concepts of polyomino, pseudo-polyomino and quasi-polyomino based on a chessboard. A polyomino or n -omino is a set of n squares from the board connected by their edges, like the movement of a rook (rook-wise connected). Similarly, a pseudo-polyomino is a set of squares from the board connected by their edges or vertices, like the movement of a queen (queen-wise connected). A quasi-polyomino is simply a set of squares from the board. Figure 14 illustrates (a) a polyomino, (b) a pseudo-polyomino and (c) a quasi-polyomino. Note that every polyomino is a pseudo-polyomino and every pseudo-polyomino is a quasi-polyomino.

Figure 14 – A polyomino, a pseudo-polyomino and a quasi-polyomino.

Source: Elaborated by the author.

Quasi-polyomino cutting and packing problems have many applications, e.g., sheet metal stamping, design of printed circuit boards, timber cutting and page layout of newspaper ([KASHKOUSH; SHALABY; ABDELHAFIEZ, 2012](#)).

The quasi-polyomino strip packing problem consists in cutting a set of items (quasi-polyominoes) from a strip with fixed width and infinite length. The objective is to minimize the length required to cut all items. According to the typology proposed by [Wäscher, Haußner and Schumann \(2007\)](#), the problem is two-dimensional with one open dimension and irregular shapes.

The quasi-polyomino strip packing problem is NP-hard, since the two-dimensional strip packing problem ([LODI; MARTELLO; MONACI, 2002](#)) can be reduced to it, using rectangular items with integer dimensions.

Several heuristic methods have been proposed for the irregular strip packing problem via an approximation of irregular shapes by quasi-polyominoes. [Pargas and Jain \(1993\)](#) developed a parallel stochastic optimization algorithm, while [Babu and Babu \(2001\)](#), [Fischer and Dagli \(2004\)](#), [Wong et al. \(2009\)](#), [Zhang and Yang \(2009\)](#), [Siasos and Vosniakos \(2014\)](#) developed genetic algorithms for the irregular strip packing problem using approximations by polyominoes. In general, these algorithms have the same basic structure. Each individual of the population is represented by a sequence of items and eventual rotations and/or flips, which is used as input for a constructive heuristic, usually the bottom-left algorithm. [Shalaby and Kashkoush \(2013\)](#) proposed a particle swarm optimization algorithm for the irregular strip packing problem. The algorithm had two stages, the first used an approximation by polyominoes to place the items and the second performed a compaction heuristic based on direct trigonometry. [Baldacci et al. \(2014\)](#) employed a lagrangian heuristic based on an approximation by polyominoes to solve real problem instances from leather cutting.

On the other hand, few exact approaches have been proposed for the problem. [Kashkoush, Shalaby and Abdelhafiez \(2012\)](#) developed a mixed integer programming model for the polyomino strip packing problem, considering one-sided polyomino (polyomino without flips) with rotations of 0 and 180 degrees. However, the objective function of their model must be modified to correctly represent the problem. In this sense, here we propose two integer programming models for the quasi-polyomino strip packing problem allowing rotations and flips of the items. Both models are based on ([BEASLEY, 1985](#); [KASHKOUSH; SHALABY; ABDELHAFIEZ, 2012](#)).

The remainder of this chapter is organized as follows. In order to make this text self-contained, in Sections 4.2 and 4.3, the models of [Beasley \(1985\)](#) and [Kashkoush, Shalaby and Abdelhafiez \(2012\)](#) are described. The two proposed models are presented in Section 4.4. In Section 4.5, computational experiments for both models are reported. Finally, Section 4.6 draws some conclusions and suggest further works.

4.2 Beasley's model

Beasley (1985) proposed a mixed integer programming model for the two-dimensional knapsack with rectangles. Given a rectangular bin with length L and width W and a set of m rectangular items such that each item i have length ℓ_i , width w_i and a value v_i . Let q_i^{MIN} and q_i^{MAX} be the minimum and maximum quantities of each item i that must be cut from the bin. Furthermore, we assume that the dimension of the rectangles are integers, the cuts are orthogonal and the orientation of the items is fixed.

To represent the placement of the items in the bin, we use the bottomleft corner as the reference point of the item (Figure 15). Thus, we can check if a cell of the bin is covered by an item using the following coefficients:

$$a_{ipqrs} = \begin{cases} 1, & \text{if the item of type } i \text{ is placed in the reference point of cell } (p, q) \\ & \text{and cover cell}(r, s); \\ 0, & \text{otherwise.} \end{cases} \quad (4.1)$$

Figure 15 – Placement of an item in the bin using Beasley (1985) model.

Source: Beasley (1985) p.51.

Formally, we can define such coefficients as:

$$a_{ipqrs} = \begin{cases} 1, & \text{if } (0 \leq p \leq r \leq p + \ell_i - 1 \leq L - 1) \wedge (0 \leq q \leq s \leq q + w_i - 1 \leq W - 1); \\ 0, & \text{otherwise.} \end{cases} \quad (4.2)$$

Let $\mathcal{L} = \{0, 1, \dots, L - 1\}$ and $\mathcal{W} = \{0, 1, \dots, W - 1\}$, respectively, be the set of placement of reference point in relation to length and width. The binary decision variable is defined as following:

$$x_{ipq} = \begin{cases} 1, & \text{if an item of type } i \text{ is place in cell } (p, q) \\ 0, & \text{otherwise.} \end{cases} \quad (4.3)$$

The model proposed by [Beasley \(1985\)](#) is given by:

$$\max \sum_{i=1}^m \sum_{p \in \mathcal{L}} \sum_{q \in \mathcal{W}} v_i x_{ipq} \quad (4.4)$$

$$\text{s.a: } \sum_{i=1}^m \sum_{p \in \mathcal{L}} \sum_{q \in \mathcal{W}} a_{ipqrs} x_{ipq} \leq 1, \quad \forall r \in \mathcal{L}, \forall s \in \mathcal{W}, \quad (4.5)$$

$$q_i^{MIN} \leq \sum_{p \in \mathcal{L}} \sum_{q \in \mathcal{W}} x_{ipq} \leq q_i^{MAX}, \quad i = 1, \dots, m, \quad (4.6)$$

$$x_{ipq} \in \{0, 1\}, \quad i = 1, \dots, m, \forall p \in \mathcal{L}, \forall q \in \mathcal{W}. \quad (4.7)$$

Objective function (4.4) maximize the value of the cut items. Constraints (4.5) prevent overlap among items. Constraints (4.6) impose the number of items to be cut is inside the allowed demand. Constraints (4.7) define the variables domains.

4.3 Kashkoush, Shalaby and Abdelhafiez's model

[Kashkoush, Shalaby and Abdelhafiez \(2012\)](#) proposed a mixed integer programming model for the polyomino tiling and strip packing problems. In the model, the strip is represented by a grid and the items are polyominos. In this sense, the problem is reduced to the placement of polyominos into a finite set of cell of the strip. The objective is to minimize the length used of the strip, placing all items without overlap.

We describe the parameters of the model as follows. Given a set of N items and a strip of width W and length L (upper bound), we represent the width and the length of each item k by w_k and ℓ_k , respectively. In this sense, $U_k = W - w_k + 1$ and $V_k = L - \ell_k + 1$ represent, respectively, the most distant position in relation to width and length where item k can be place in the strip. To represent each polyomino k we use a binary matrix P_k , where each position P_{kmn} indicates if cell (m, n) of polyomino k is occupied in the original orientation (rotation 0°), since the model allow that items rotate 180° (but it does not allow flip). The topleft corner of each item is used as the reference point to place the polyomino in the strip.

The binary decisions of the model are:

$$x_{kij} = \begin{cases} 1, & \text{if polyomino } k \text{ with rotation } 0^\circ \text{ is placed in cell } (i, j); \\ 0, & \text{otherwise.} \end{cases} \quad (4.8)$$

$$y_{kij} = \begin{cases} 1, & \text{if polyomino } k \text{ with rotation } 180^\circ \text{ is placed in cell } (i, j); \\ 0, & \text{otherwise.} \end{cases} \quad (4.9)$$

$$s_{kijmn} = \begin{cases} 1, & \text{if polyomino } k \text{ is placed in cell } (i, j) \text{ and cover cell } (m, n); \\ 0, & \text{otherwise.} \end{cases} \quad (4.10)$$

The model proposed by Kashkoush, Shalaby and Abdelhafiez (2012) is given by:

$$\max \sum_{k=1}^N \sum_{i=1}^{U_k} \sum_{j=1}^{V_k} \sum_{m=1}^W \sum_{n=1}^L n s_{kijmn} \quad (4.11)$$

$$\begin{aligned} & k = 1, \dots, N, \\ & i = 1, \dots, U_k, \\ \text{s.a: } & s_{kij(m+i-1)(n+j-1)} - P_{kmn} x_{kij} - P_{k(w_k-m+1)(l_k-n+1)} y_{kij} = 0, \quad j = 1, \dots, V_k, \\ & m = 1, \dots, w_k, \\ & n = 1, \dots, l_k, \end{aligned} \quad (4.12)$$

$$\sum_{i=1}^{U_k} \sum_{j=1}^{V_k} x_{kij} + \sum_{i=1}^{U_k} \sum_{j=1}^{V_k} y_{kij} = 1, \quad k = 1, \dots, N, \quad (4.13)$$

$$\sum_{k=1}^N \sum_{i=1}^{U_k} \sum_{j=1}^{V_k} s_{kijmn} \leq 1, \quad \begin{aligned} & m = 1, \dots, W, \\ & n = 1, \dots, L, \end{aligned} \quad (4.14)$$

$$\begin{aligned} & x_{kij} \in \{0, 1\}, \\ & k = 1, \dots, N, \\ & i = 1, \dots, U_k, \\ & j = 1, \dots, V_k, \end{aligned} \quad (4.15)$$

$$\begin{aligned} & y_{kij} \in \{0, 1\}, \\ & k = 1, \dots, N, \\ & i = 1, \dots, U_k, \\ & j = 1, \dots, V_k, \end{aligned} \quad (4.16)$$

$$\begin{aligned} & s_{kijmn} \geq 0, \\ & k = 1, \dots, N, \\ & i = 1, \dots, U_k, \\ & j = 1, \dots, V_k \\ & m = 1, \dots, w_k, \\ & n = 1, \dots, l_k. \end{aligned} \quad (4.17)$$

Objective function (4.11) penalize the usage of each cell, gradually increasing the weight for more distant cells, according to Figure 16. However, we should highlight that even the objective function penalize the more distant columns, it does not impose that the length used to place all the items in the strip is minimized. It is possible to obtain a counter-example for objective function (Figure 17), what shows that the model does not precisely represent the strip packing problem.

Constraints (4.12) associate auxiliary variables to the placement variables in the strip. Constraints (4.13) impose that only one rotation of each item is used (0° or 180°). Constraints (4.14) prevent overlap among items, allowing at most one item occupy each cell of the strip. Constraints (4.15), (4.16) and (4.17) define the variable domains.

Figure 16 – Calculation of objective function in the model of [Kashkoush, Shalaby and Abdelhafiez \(2012\)](#).
(a) Alocated items. (b) Penalties.

1	1	0	0	1	0	0	1
0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	0
1	1	1	1	1	1	0	0

(a)

1	2	0	0	5	0	0	8
0	2	3	4	5	6	7	8
0	2	3	4	5	6	7	0
1	2	3	4	5	6	0	0

(b)

Source: [Kashkoush, Shalaby and Abdelhafiez \(2012\)](#) p.396.

Figure 17 – Example of solutions for the problem that minimize the penalties of used cells. (a) Solution that minimize the penalties, but, it does not minimize the length. (b) Solution has a higher penalty, but it has a smaller length.

0	2	0	0
1	2	0	0
1	2	0	0
0	2	3	4

(a) Penalty = 17
Length = 4

0	0	3
0	2	3
0	2	3
1	2	3

(b) Penalty = 19
Length = 3

Source: Elaborated by the author.

4.4 Proposed model

The quasi-polyomino strip packing problem consists in cutting a set of items that are represented by quasi-polyominoes from a rectangular strip with fixed width and infinite length. The objective is to cut all items such that the length of the strip is minimized. The items cannot overlap, but they can be rotated (0° , 90° , 180° e 270°) and/or flipped around one of their axes.

Figure 18 – A quasi-polyomino strip packing problem. (a) Items. (b) Solution.

Source: Elaborated by the author.

In this section we propose two models for the studied problem, but first we describe how the items and the strip are represented.

4.4.1 Representing the items and the strip

To prevent symmetrical solutions, the items that can be represented by the same quasi-polyomino are treated as an item type. In this sense, each item type $t = 1, \dots, T$ is described by a binary matrix $A^t = [a_{ij}^t]$ of dimensions $w_t \times \ell_t$ (width $\times$ length). Similarly, the strip is represented by a matrix of dimensions $W \times L$. Since the strip has one open dimension, we need to estimate the value of L , that defines an upper bound for the length of the strip for both models.

To represent the operations of rotation and/or flip of quasi-polyominoes, we use multiplications by permutation matrices (identity I and/or exchange J) and transpositions of the binary matrix that represents the item, where:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad J = \begin{bmatrix} 0 & 0 & 1 \\ 0 & \ddots & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Figure 19 – Representing rotation and flip of a quasi-polyomino.

Source: Elaborated by the author.

Thus, each rotation/flip $r = 1, \dots, R_t$ of an item type t is described by a binary matrix $B^{tr} = [b_{ij}^{tr}]$ of dimensions $w_{tr} \times \ell_{tr}$, such that $B^{tr} = P^{tr} A^t Q^{tr}$ or $B^{tr} = P^{tr} (A^t)^T Q^{tr}$, where $P^{tr}, Q^{tr} \in \{I, J\}$. For example, Figure 19 (a) represents an item without rotations while Figure 19 (b) represents the 90° rotation of the item and it is obtained by a multiplication $B^{tr} = J(A^{tr})^T I$. Similarly, Figure 19 (e) represents the flip of the item by the multiplication $B^{tr} = JA^{tr} I$.

4.4.2 Problem constraints

The problem has two basic sets of constraints: demand constraints and non-overlap constraints.

4.4.2.1 Demand constraints

To place the items in the strip, we use its left upper corner as its reference point. Let $(0,0)$ be the origin at the leftmost topmost position of the strip. We define $W_{tr} = W - w_{tr} + 1$ as the bottommost cell and $L_{tr} = L - \ell_{tr} + 1$ as the rightmost cell where the reference point of item type t in rotation/flip r can be placed in the strip (see Figure 20). Given an item type t in rotation/flip r , we define a binary decision variable x_{ij}^{tr} for each admissible cell (i, j) of the strip, such that $i = 1, \dots, W_{tr}$ and $j = 1, \dots, L_{tr}$:

$$x_{ij}^{tr} = \begin{cases} 1, & \text{if the reference point of a quasi-polyomino of type } t \text{ in rotation/} \\ & \text{flip } r \text{ is placed on cell } (i, j) \text{ of the strip;} \\ 0, & \text{otherwise.} \end{cases} \quad (4.18)$$

Thus, constraints (4.19) ensure the demand q_t for each item type t is met and impose all items are entirely inside the strip.

$$\sum_{r=1}^{R_t} \sum_{i=1}^{W_{tr}} \sum_{j=1}^{L_{tr}} x_{ij}^{tr} = q_t, \quad t = 1, \dots, T. \quad (4.19)$$

4.4.2.2 Non-overlap constraints

To prevent overlap among items, we must check if there is more than one item type covering the same cell of the strip, as described by constraints (4.20). Therefore, given a cell (m, n) of the strip, we analyze every binary variable x_{ij}^{tr} that can cover it. Thereby, given an item type t in a rotation/flip r , we need to analyze only variables x_{ij}^{tr} in the rectangle defined by the set $\{(i, j) \mid i = m - w_{tr} + 1, \dots, m; j = n - \ell_{tr} + 1, \dots, n\}$. To eliminate the sum of cells outside the strip, we use the minimum and maximum in the index of the sum.

$$\sum_{t=1}^T \sum_{r=1}^{R_t} \sum_{i=\max(1, m-w_{tr}+1)}^{\min(m, W_{tr})} \sum_{j=\max(1, n-\ell_{tr}+1)}^{\min(n, L_{tr})} b_{m-i+1, n-j+1}^{tr} x_{ij}^{tr} \leq 1, \quad \begin{matrix} m = 1, \dots, W, \\ n = 1, \dots, L. \end{matrix} \quad (4.20)$$

Figure 20 illustrates the region that must be analyzed to check overlap in a cell (m, n) of the strip. Note an item type t in rotation/flip r placed in cell (i, j) covers the cell (m, n) if $x_{ij}^{tr} = 1$ and $b_{m-i+1, n-j+1}^{tr} = 1$.

Figure 20 – Overlap verification and objective function calculation for a cell of the strip.

Source: Elaborated by the author.

Next, we present the specific characteristics of the two proposed models for the problem.

4.4.3 Strip_Max model

The first model, denoted by Strip_Max, uses an integer variable z to define the length used of the strip and is given by:

$$\min z \quad (4.21)$$

$$\text{s.t. } n \sum_{t=1}^T \sum_{r=1}^{R_t} \sum_{i=\max(1, m-w_{tr}+1)}^{\min(m, W_{tr})} \sum_{j=\max(1, n-\ell_{tr}+1)}^{\min(n, L_{tr})} b_{m-i+1, n-j+1}^{tr} x_{ij}^{tr} \leq z, \quad \begin{matrix} m = 1, \dots, W, \\ n = 1, \dots, L, \end{matrix} \quad (4.22)$$

$$\sum_{r=1}^{R_t} \sum_{i=1}^{W_{tr}} \sum_{j=1}^{L_{tr}} x_{ij}^{tr} = q_t, \quad t = 1, \dots, T, \quad (4.19)$$

$$\sum_{t=1}^T \sum_{r=1}^{R_t} \sum_{i=\max(1, m-w_{tr}+1)}^{\min(m, W_{tr})} \sum_{j=\max(1, n-\ell_{tr}+1)}^{\min(n, L_{tr})} b_{m-i+1, n-j+1}^{tr} x_{ij}^{tr} \leq 1, \quad \begin{matrix} m = 1, \dots, W, \\ n = 1, \dots, L, \end{matrix} \quad (4.20)$$

$$x_{ij}^{tr} \in \{0, 1\}, \quad \begin{matrix} t = 1, \dots, T, \\ r = 1, \dots, R_t, \\ i = 1, \dots, W_{tr}, \\ j = 1, \dots, L_{tr}, \end{matrix} \quad (4.23)$$

$$z \in \mathbb{Z}. \quad (4.24)$$

Objective function (4.21) together with constraints (4.22) minimize the length used to place all items on the strip. Constraints (4.22) are based on constraints (4.20) and ensure the length used of the strip is greater than or equal to the column index of each cell covered by an item (see Figure 20). Constraints (4.19) impose the demands are met for each item type. Constraints (4.20) ensure each cell of the strip is covered by at most one item type. Constraints (4.23) and (4.24) define the domain of the variables.

4.4.4 Strip_Sum model

The second model, denoted by Strip_Sum, uses binary variables ϕ_n to denote if the n -th column of the strip has any cell covered by an item, i.e.:

$$\phi_n = \begin{cases} 1, & \text{if column } n \text{ of the strip has any cell covered by an item;} \\ 0, & \text{otherwise.} \end{cases} \quad (4.25)$$

The Strip_Sum model is given by:

$$\min \sum_{n=1}^L \phi_n \quad (4.26)$$

$$\text{s.t. } \sum_{t=1}^T \sum_{r=1}^{R_t} \sum_{i=\max(1, m-w_{tr}+1)}^{\min(m, W_{tr})} \sum_{j=\max(1, n-\ell_{tr}+1)}^{\min(n, L_{tr})} b_{m-i+1, n-j+1}^{tr} x_{ij}^{tr} \leq \phi_n, \quad \begin{matrix} m = 1, \dots, W, \\ n = 1, \dots, L, \end{matrix} \quad (4.27)$$

$$\sum_{r=1}^{R_t} \sum_{i=1}^{W_{tr}} \sum_{j=1}^{L_{tr}} x_{ij}^{tr} = q_t, \quad t = 1, \dots, T, \quad (4.19)$$

$$\phi_n \geq \phi_{n+1}, \quad n = 1, \dots, L-1, \quad (4.28)$$

$$x_{ij}^{tr} \in \{0, 1\}, \quad \begin{matrix} t = 1, \dots, T, \\ r = 1, \dots, R_t, \\ i = 1, \dots, W_{tr}, \\ j = 1, \dots, L_{tr}, \end{matrix} \quad (4.23)$$

$$\phi_n \in \{0, 1\}, \quad n = 1, \dots, L. \quad (4.29)$$

Objective function (4.26) minimizes the number of columns used to place all items on the strip. Constraints (4.27) are based on constraints (4.20) and ensure a cell is covered only if its respective column is used. Furthermore, they prevent the overlap among items (see Figure 20). Constraints (4.28) avoid the use of non-consecutive columns, which ensures the objective function minimizes the length used of the strip. Note that ϕ_n could be 1 even if column n is not covered but one of the next is covered, since a quasi-polyomino is not necessarily a connected pattern of squares. Constraints (4.19) and (4.23) are the same than the previous model, while constraints (4.29) define the domain of the variables associated to the activation of the columns.

4.4.5 Analysis of the models

Table 8 presents a summary about the number of constraints and binary and real variables of the two proposed models. For both models, the number of constraints is linear in the number of item types and cells of the strip. The number of binary variable is proportional to the sum of the number of admissible cells for each item type in the strip and eventually to the number of columns, while the number of integer variables is at most one.

Table 8 – Number of constraints and variables of the proposed models.

Model	Constraints	Variables	
		Binary	Integer
Strip_Max	$T + 2WL$	$\sum_{t=1}^T \sum_{r=1}^{R_t} W_{tr} L_{tr}$	1
Strip_Sum	$T + WL + L - 1$	$\sum_{t=1}^T \sum_{r=1}^{R_t} W_{tr} L_{tr} + L$	0

4.4.6 Valid inequalities and variable fixing

To enhance both models, we can use the valid inequalities proposed by [Rodrigues \(2015\)](#) in the Strip_Max model, for both demand (4.19) and non-overlap constraints (4.20). We can obtain the value of the objective function using constraints (4.30), that define the length used based on the reference column summed with the length associated to each item type.

$$(j + \ell_{tr} - 1)x_{ij}^{tr} \leq z, \quad t = 1, \dots, T, r = 1, \dots, R_t, i = 1, \dots, W_{tr}, j = 1, \dots, L_{tr}. \quad (4.30)$$

Summing all constraints (4.30) for an item type and using constraints (4.19), we obtain the valid inequalities (4.31).

$$\sum_{r=1}^{R_t} \sum_{i=1}^{W_{tr}} \sum_{j=1}^{L_{tr}} (j + \ell_{tr} - 1)x_{ij}^{tr} \leq q_t z, \quad t = 1, \dots, T \quad (4.31)$$

Let $\underline{z}$ be a lower bound for the length used of the strip. We can rewrite constraints (4.31) to obtain inequalities (4.32).

$$q_t \underline{z} + \sum_{r=1}^{R_t} \sum_{i=1}^{W_{tr}} \sum_{j=\max(1, \underline{z} - \ell_{tr} + 2)}^{L_{tr}} (j + \ell_{tr} - 1 - \underline{z})x_{ij}^{tr} \leq q_t z, \quad t = 1, \dots, T. \quad (4.32)$$

Similarly, we can use constraints (4.20) to generate inequalities (4.33).

$$\underline{z} + (n - \underline{z}) \sum_{t=1}^T \sum_{r=1}^{R_t} \sum_{i=\max(1, m - w_{tr} + 1)}^{\min(m, W_{tr})} \sum_{j=\max(1, n - \ell_{tr} + 1)}^{\min(n, L_{tr})} b_{m-i+1, n-j+1}^{tr} x_{ij}^{tr} \leq z, \quad \begin{matrix} m = 1, \dots, W, \\ n = \underline{z} + 1, \dots, L. \end{matrix} \quad (4.33)$$

4.4.7 Strip_Sum model

On the other hand, for the Strip_Sum model we can fix all variables ϕ_n that are associated to any column with value less than or equal to the lower bound $\lceil \underline{z} \rceil$, as is described by equations (4.34).

$$\phi_n = 1, \quad n = 1, \dots, \lceil \underline{z} \rceil. \quad (4.34)$$

Similarly to the Strip_Max model, we can rewrite valid inequalities (4.32) for the Strip_Sum model to generate the inequalities (4.35).

$$q_t \underline{z} + \sum_{r=1}^{R_t} \sum_{i=1}^{W_{tr}} \sum_{j=\max(1, \underline{z}-\ell_{tr}+2)}^{L_{tr}} (j + \ell_{tr} - 1 - \underline{z}) x_{ij}^{tr} \leq q_t \sum_{n=1}^L \phi_n, \quad t = 1, \dots, T. \quad (4.35)$$

4.5 Computational experiments

In this section, we describe the instances and the results obtained by Strip_Sum and Kashkoush, Shalaby and Abdelhafiez (2012) models. Since Strip_Max results are similar to Strip_Sum, they are not shown. The experiments were conducted in a computer with Intel Core i7-2600 @3.40 GHz processor and 16 GB of RAM under the Ubuntu 14.04 LTS operating system. To solve both models, we used the commercial solver IBM ILOG CPLEX 12.6 with default parameter and a time limit of 3,600 seconds.

In the Strip_Sum model it was fixed some variables (4.34). The lower bound used was the ceil of the sum of the area of all items divided by the width of the strip, i. e.:

$$\underline{z} = \left\lceil \frac{1}{W} \sum_{t=1}^T q_t \sum_{i=1}^{w_t} \sum_{j=1}^{\ell_t} a_{ij}^t \right\rceil. \quad (4.36)$$

4.5.1 Instances

The two models were evaluated using two set of instances. The first one has 53 instances taken or adapted from the literature without rotations and flips of cutting and packing problems of polyominoes or irregular items. The second one is composed by a set of 60 random instances divided into four groups with and without rotation and flips.

4.5.1.1 Instance set 1

The _poly instances are approximation via polyomino of classical irregular strip packing problems, using unit discretization. The characteristics of the instances evaluated are presented in Table 9. The first column describes the instance name, while the second column describes its source. The third and fourth columns indicates, respectively, the quantity of items and item types. The following two columns present the initial length (upper bound) and the width of the strip.

4.5.1.2 Instance set 2

Each instance is describe by: *group_pieces_demand_width_minDim_maxDim_inst*, where:

- group - A, B, C, D represent the densities of pieces 0.25, 0.5, 0.75 and 0.25 - 0.75

$$density = \frac{\sum_{i=1}^w \sum_{j=1}^{\ell} a_{ij}}{\ell w};$$

- pieces - represent the number of pieces T ;
- demand - represent the demand of each piece q_t ;
- width - represent the strip width W ;
- minDim - represent the minimum length/width of the pieces ℓ/w ;
- maxDim - represent the maximum length/width of the pieces ℓ/w ;
- inst - a, b, c, d, e represent one instance of a group (each group has five instances).

4.5.2 Computational Results

4.5.2.1 Instance set 1

The results of the computational experiments for [Kashkoush, Shalaby and Abdelhafiez \(2012\)](#) and Strip_Sum models for literature instances are presented in Figure 21 and Table 10. For each model, the figure shows the number of binaries, variables and constraints. Note that ([KASHKOUSH; SHALABY; ABDELHAFIEZ, 2012](#)) model has more binaries, variables and constraints than Strip_Sum model. For each model and instance, the table reports the value of the objective function (Obj), Length, optimality gap (in %) and runtime (in seconds).

Table 9 – Features of the evaluated instances.

Instance	Source	P	T	L	W
Takefuji	Takefuji and Lee (1990)	10	10	9	7
Asai1	Asai, Nakayama and Ninomiya (1996)	11	11	10	8
Asai2	Asai, Nakayama and Ninomiya (1996)	12	12	10	8
Asai3	Asai, Nakayama and Ninomiya (1996)	12	12	11	9
Asai4	Asai, Nakayama and Ninomiya (1996)	14	14	13	10
Amintoosi	Amintoosi <i>et al.</i> (2007)	10	10	21	15
Manabe1	Manabe and Asai (1999)	5	5	5	5
Manabe2	Manabe and Asai (1999)	5	5	5	5
Manabe3	Manabe and Asai (1999)	5	5	5	5
Manabe4	Manabe and Asai (1999)	6	6	6	6
Gwee1	Gwee and Lim (1996)	8	8	8	8
Gwee2	Gwee and Lim (1996)	10	10	9	7
Gwee3	Gwee and Lim (1996)	12	12	12	6
Gwee4	Gwee and Lim (1996)	128	8	39	32
Matayoshi1-1	Matayoshi (2011)	8	8	24	20
Matayoshi1-2	Matayoshi (2011)	8	7	20	20
Matayoshi1-3	Matayoshi (2011)	9	8	23	20
Matayoshi1-4	Matayoshi (2011)	9	9	23	20
Matayoshi1-5	Matayoshi (2011)	10	9	23	20
Matayoshi2-1	Matayoshi (2011)	5	5	22	19
Matayoshi2-2	Matayoshi (2011)	7	7	22	19
Matayoshi2-3	Matayoshi (2011)	7	7	22	19
Matayoshi2-4	Matayoshi (2011)	8	8	22	19
Matayoshi2-5	Matayoshi (2011)	10	10	28	19
Babu	Babu and Babu (2001)	14	3	4	8
Siasos	Siasos and Vosniakos (2014)	15	7	7	10
Jain1	Jain and Gea (1998)	26	1	12	10
Jain2	Jain and Gea (1998)	19	1	11	10
Jain3	Jain and Gea (1998)	20	1	22	24
Jain4	Jain and Gea (1998)	42	3	12	12
Jain5	Jain and Gea (1998)	40	11	15	21
Jain6	Jain and Gea (1998)	100	9	45	51
Fujiyoshi1	Fujiyoshi and Murata (2000)	12	12	10	7
Fujiyoshi2	Fujiyoshi and Murata (2000)	19	18	18	26
Lin1	Lin, Chen and Chang (2002)	17	12	9	8
Lin2	Lin, Chen and Chang (2002)	29	17	13	12
RL10	Wu, Chang and Chang (2000)	10	8	11	10
RLT30	Wu, Chang and Chang (2000)	30	27	36	30
HHIY10_10	Hu <i>et al.</i> (2015)	10	10	11	9
HHIY10_100	Hu <i>et al.</i> (2015)	100	10	34	29
HHIY100_100	Hu <i>et al.</i> (2015)	100	98	41	32
T144_001	Hu <i>et al.</i> (2015)	20	20	13	11
T144_002	Hu <i>et al.</i> (2015)	40	20	18	15
T144_004	Hu <i>et al.</i> (2015)	80	20	25	22
T144_008	Hu <i>et al.</i> (2015)	160	20	34	31
T144_016	Hu <i>et al.</i> (2015)	320	20	48	44
T144_032	Hu <i>et al.</i> (2015)	640	20	67	62
Blasz1_poly	Błażewicz, Hawryluk and Walkowiak (1993)	28	7	61	30
Blasz2_poly	Błażewicz, Hawryluk and Walkowiak (1993)	20	4	54	30
Fu_poly	Fujita, Akagi and Hirokawa (1993)	12	11	38	38
Jakobs1_poly	Jakobs (1996)	25	21	13	40
Jakobs2_poly	Jakobs (1996)	25	22	29	70
Shapes0_poly	Oliveira and Ferreira (1993)	43	4	67	40

Table 10 – Results obtained for the strip packing problem.

Instance	(KASHKOUSH; SHALABY; ABDELHAFIEZ, 2012)				Strip_Sum		
	Obj	Length	Gap (%)	Time(s)	Obj / Length	Gap (%)	Time (s)
Takefuji	196	7	0.00	0.27	7	0.00	0.00
Asai1	288	8	0.00	0.22	8	0.00	0.00
Asai2	288	8	0.00	0.26	8	0.00	0.00
Asai3	405	9	0.00	0.07	9	0.00	0.00
Asai4	550	10	0.00	0.19	10	0.00	0.00
Amintoosi	3465	21	0.00	0.50	21	0.00	0.00
Manabe1	75	5	0.00	0.00	5	0.00	0.00
Manabe2	75	5	0.00	0.00	5	0.00	0.00
Manabe3	75	5	0.00	0.00	5	0.00	0.00
Manabe4	126	6	0.00	0.01	6	0.00	0.01
Gwee1	288	8	0.00	0.03	8	0.00	0.01
Gwee2	196	7	0.00	0.07	7	0.00	0.00
Gwee3	335	12	0.00	0.54	11	0.00	0.41
Gwee4	-	-	-	ML	36	8.33	3600.00
Matayoshi1-1	4200	20	0.00	1.51	20	0.00	0.33
Matayoshi1-2	4200	20	0.00	0.38	20	0.00	0.06
Matayoshi1-3	4200	20	0.00	0.85	20	0.00	0.32
Matayoshi1-4	4200	20	0.00	0.83	20	0.00	0.43
Matayoshi1-5	4200	20	0.00	0.84	20	0.00	0.30
Matayoshi2-1	4807	22	0.00	0.21	22	0.00	0.06
Matayoshi2-2	4807	22	0.00	0.32	22	0.00	0.00
Matayoshi2-3	4807	22	0.00	0.28	22	0.00	0.09
Matayoshi2-4	4807	22	0.00	0.30	22	0.00	0.07
Matayoshi2-5	4807	22	0.00	1.16	22	0.00	0.22
Babu	80	4	0.00	0.02	4	0.00	0.00
Siasos	213	7	0.00	0.75	7	0.00	0.05
Jain1	401	10	0.00	0.40	10	0.00	0.04
Jain2	392	11	0.00	0.13	10	0.00	0.03
Jain3	3636	20	0.00	2.49	20	0.00	0.00
Jain4	651	11	0.00	4.84	11	0.00	0.42
Jain5	2325	15	0.00	5.85	15	0.00	1.17
Jain6	-	-	-	ML	-	-	3600.00
Fujiyoshi1	295	9	0.00	0.78	9	0.00	0.14
Fujiyoshi2	2599	16	8.74	3,600.00	16	12.50	3600.00
Lin1	288	8	0.00	0.40	8	0.00	0.18
Lin2	936	12	0.00	1.56	12	0.00	0.47
RL10	555	11	0.00	0.13	11	0.00	0.04
RLT30	14163	33	1.50	3,600.00	33	9.09	3600.00
HHIY10_10	436	10	0.00	0.36	10	0.00	0.25
HHIY10_100	-	-	-	ML	32	12.50	3600.00
HHIY100_100	-	-	-	ML	40	20.00	3600.00
T144_001	738	12	0.00	2.97	12	0.00	0.53
T144_002	2108	17	0.00	25.00	17	0.00	1.43
T144_004	5658	23	0.00	1,818.43	23	0.00	6.95
T144_008	-	-	-	ML	32	0.00	183.23
T144_016	-	-	-	ML	45	0.00	2267.05
T144_032	-	-	-	ML	64	1.56	3600.00
Blasz1_poly	-	-	-	ML	59	12.64	3600.00
Blasz2_poly	33368	54	3.80	3,600.00	54	5.56	3600.00
Fu_poly	18864	35	0.01	3,193.56	35	8.57	3600.00
Jakobs1_poly	2489	13	2.33	3,600.00	12	8.33	3600.00
Jakobs2_poly	-	-	-	ML	28	21.43	3600.00
Shapes0_poly	-	-	-	ML	-	-	3600.00
#Feasible	43				51		
#Optimal	38				40		

Figure 21 – Number of binaries, variables and constraints for Kashkoush et al. and Strip Sum models (in logscale).

Source: Elaborated by the author.

Strip_Sum model obtained better or equal results than (KASHKOUSH; SHALABY; ABDELHAFIEZ, 2012) model for all instances in relation to length, gap and time. For eight instances only Strip_Sum model obtained feasible solutions and two of the optimal solution were obtained. The proposed model obtained an optimal solution for instances with up to 320 items (T144_016). The model proved optimality for 40 out 53 of the evaluated instances. The best solutions obtained by the Strip_Sum model can be seen in the Appendix C

4.5.2.2 Instance set 2

We run tests for random instances with and without rotations. Results for instances without rotations and flips are shown in Table 11 and 13, while results for instances with rotations and flips are shown in Table 12 and 14. Each class of instance describe a group of five instances. For each class tables describe the average number of binaries (Avg. bin.), average number of constraints (avg. cons.), number of optimal solutions (#Opt.), average number of upper bound (Avg. UB), average objective function (Avg. Obj.), average optimality gap (Avg. gap) and average time (avg. time). As expected, we found more optimal solutions when there are no rotation and flips. Moreover, we found more optimal solutions for class A than other classes, because they have lower density. Furthermore, as expected, when the dimension of the pieces increases we cannot prove optimality and the average gap increases.

4.6 Conclusions and further work

We have developed two integer programming models the quasi-polyomino strip packing problem based on (BEASLEY, 1985) and (KASHKOUSH; SHALABY; ABDELHAFIEZ, 2012). Strip_Sum model proved optimality for 40 out 53 of the evaluated literature instances. Furthermore, it can solve instances with up to 320 items of 20 item types to optimality. For further work, we suggest to develop a branch-and-cut solver using the valid inequalities described in this chapter.

Table 11 – Results obtained for the strip packing problem - random instances without rotations and flips (part 1).

Class	Avg. Bin.	Avg. Cons.	#Opt.	Avg. UB	Avg. Obj.	Avg. Gap(%)	Avg. Time(s)
A_10_1_20_3_5	368.4	124.0	5	5.0	5.0	0.0	0.0
A_10_1_20_5_7	649.2	212.2	5	9.2	8.2	0.0	2.8
A_10_1_20_7_10	1,900.0	493.6	3	22.6	20.0	5.5	2,726.9
B_10_1_20_3_5	717.4	170.2	5	7.2	6.4	0.0	7.3
B_10_1_20_5_7	1,754.2	367.6	4	16.6	15.6	1.3	1,324.4
B_10_1_20_7_10	3,971.0	854.8	1	39.8	36.8	3.2	3,197.9
C_10_1_20_3_5	954.6	199.6	5	8.6	8.4	0.0	10.7
C_10_1_20_5_7	2,282.6	443.2	5	20.2	19.6	0.0	61.1
C_10_1_20_7_10	4,229.0	875.8	5	40.8	40.4	0.0	402.4
D_10_1_20_3_5	638.2	157.6	5	6.6	5.8	0.0	1.2
D_10_1_20_5_7	1,699.4	363.4	5	16.4	15.2	0.0	998.9
D_10_1_20_7_10	3,645.2	779.2	2	36.2	33.4	5.4	2,283.1
#Optimal	50						

Table 12 – Results obtained for the strip packing problem - random instances with rotations and flips (part 1).

Class	Avg. Bin.	Avg. Cons.	#Opt.	Avg. UB	Avg. Obj.	Avg. Gap(%)	Avg. Time(s)
A_10_1_20_3_5	2,355.2	115.6	5	4.6	4.6	0.0	0.0
A_10_1_20_5_7	4,657.8	203.8	5	8.8	6.6	0.0	6.0
A_10_1_20_7_10	14,611.2	485.2	0	22.2	19.0	36.4	3,600.0
B_10_1_20_3_5	5,373.6	166.0	5	7.0	5.2	0.0	312.4
B_10_1_20_5_7	14,155.4	371.8	0	16.8	14.2	26.2	3,600.0
B_10_1_20_7_10	89,342.0	846.4	0	39.4	36.4	45.3	3,600.0
C_10_1_20_3_5	7,570.0	199.6	2	8.6	7.8	7.2	2,336.2
C_10_1_20_5_7	16,450.2	413.8	0	18.8	17.8	17.6	3,600.0
C_10_1_20_7_10	32,080.2	842.2	0	39.2	38.6	27.4	3,600.0
D_10_1_20_3_5	4,464.8	149.2	5	6.2	5.0	0.0	168.6
D_10_1_20_5_7	13,210.8	359.2	0	16.2	14.2	24.2	3,600.0
D_10_1_20_7_10	28,403.0	766.6	0	35.6	33.4	41.4	3,600.0
#Optimal	22						

Table 13 – Results obtained for the strip packing problem - random instances without rotations and flips (part 2).

Class	Avg. Bin.	Avg. Cons.	#Opt.	Avg. UB	Avg. Obj.	Avg. Gap(%)	Avg. Time(s)
D_1_10_20_3_5	91.6	160.6	5	7.6	6.8	0.0	0.1
D_1_10_20_5_7	263.4	454.6	5	21.6	21.4	0.0	0.6
D_1_10_20_7_10	494.8	988.0	5	47.0	47.0	0.0	1.4
D_1_20_20_3_5	204.6	295.0	5	14.0	12.6	0.0	0.6
D_1_20_20_5_7	528.8	811.6	5	38.6	37.2	0.0	9.9
D_1_20_20_7_10	1,094.6	2,004.4	5	95.4	83.4	0.0	87.2
D_5_2_20_3_5	338.8	156.0	5	7.0	6.4	0.0	0.6
D_5_2_20_5_7	825.0	345.0	5	16.0	15.0	0.0	44.8
D_5_2_20_7_10	1,739.6	748.2	5	35.2	34.0	0.0	1,176.9
D_5_4_20_3_5	870.0	286.2	5	13.2	12.0	0.0	624.2
D_5_4_20_5_7	2,011.6	681.0	0	32.0	30.8	18.2	3,600.0
D_5_4_20_7_10	4,094.4	1,537.8	0	72.8	68.6	16.7	3,600.0
#Optimal	50						

Table 14 – Results obtained for the strip packing problem - random instances with rotations and flips (part 2).

Class	Avg. Bin.	Avg. Cons.	#Opt.	Avg. UB	Avg. Obj.	Avg. Gap(%)	Avg. Time(s)
D_1_10_20_3_5	480.2	135.4	5	6.4	5.6	0.0	1.4
D_1_10_20_5_7	1,492.4	374.8	5	17.8	15.8	0.0	33.7
D_1_10_20_5_7	2,932.2	824.2	5	39.2	35.4	0.0	286.0
D_1_20_20_3_5	1,218.4	248.8	4	11.8	10.0	1.4	727.1
D_1_20_20_5_7	3,524.8	736.0	0	35.0	31.6	15.5	3,600.0
D_1_20_20_5_7	6,753.4	1,672.6	1	79.6	73.8	16.1	2,929.8
D_5_2_20_3_5	2,543.8	151.8	4	6.8	5.8	2.5	729.7
D_5_2_20_5_7	6,207.8	336.6	0	15.6	13.0	10.5	3,600.0
D_5_2_20_7_10	13,091.8	723.0	0	34.0	31.4	35.5	3,600.0
D_5_4_20_3_5	6,480.0	273.6	0	12.6	10.6	12.9	3,600.0
D_5_4_20_5_7	15,478.6	664.2	0	31.2	28.2	34.8	3,600.0
D_5_4_20_7_10	31,537.0	1,504.2	0	71.2	68.6	49.3	3,600.0
#Optimal	24						

FINAL REMARKS AND FURTHER WORKS

In this thesis, we developed mathematical models for irregular strip packing problems. The first two models deal with the irregular strip packing problem and the third approaches the quasi-polyomino strip packing problem. The research resulted in two papers.

In the first paper, we have proposed a clique covering MIP model for the irregular strip packing problem based on the dotted-board model (TOLEDO *et al.*, 2013). It can solve instances with up to 25 pieces of 22 piece types to optimality, subject to the board discretization. Moreover, it obtained better performance than the dotted-board model for most instances and solved larger instances to optimality in comparison to state-of-the-art models. We have also introduced valid inequalities for the clique covering model and developed an iterative method to calculate a better lower bound and enhance the formulation.

In the second paper, we have proposed symmetry breaking constraints for the irregular strip packing problem. Although this new set of constraints was added to NFP-CM proposed in (CHERRI *et al.*, 2016), the ideas presented here can be applied to other models.

Computational experiments show the relevance of this new set of constraints. In general, the symmetry breaking constraints improve lower bounds and reduce run-time and the number of nodes explored to prove optimality.

Finally, we developed two integer programming models the quasi-polyomino strip packing problem based on (BEASLEY, 1985) and (KASHKOUSH; SHALABY; ABDELHAFIEZ, 2012). Strip_Sum model proved optimality for 40 out 53 of the evaluated literature instances. Furthermore, they can solve instances with up to 320 items of 20 item types to optimality.

We suggest adding classical valid inequalities from the stable set polytope (REBENACK; REINELT; PARDALOS, 2012) to improve the linear relaxation of the clique covering model. Such inequalities, as well the ones proposed here can be used in the development of a branch-and-cut algorithm for the model described in Chapter 2.

Furthermore, a promising line of research it to develop symmetry breaking constraints for instances with non-convex pieces and study branching priorities for binary variables for the model describe in Chapter 3.

Finally, we suggest developing a branch-and-cut solver using the valid inequalities described in Chapter 4 for Strip_Sum and Strip_Max models.

BIBLIOGRAPHY

ALVAREZ-VALDES, R.; MARTINEZ, A.; TAMARIT, J. A branch & bound algorithm for cutting and packing irregularly shaped pieces. **International Journal of Production Economics**, v. 145, p. 463 – 477, 2013. Citations on pages [3](#), [10](#), [21](#), [30](#), and [35](#).

AMARO JR, B.; PINHEIRO, P. R.; COELHO, P. V. A parallel biased random-key genetic algorithm with multiple populations applied to irregular strip packing problems. **Mathematical Problems in Engineering**, v. 2017, 2017. Citation on page [2](#).

AMINTOOSI, M.; YAZDI, H. S.; FATHY, M.; MONSEFI, R. Using pattern matching for tiling and packing problems. **European Journal of Operational Research**, v. 183, p. 950 – 960, 2007. Citations on pages [5](#) and [50](#).

ASAI, H.; NAKAYAMA, T.; NINOMIYA, H. Tiling algorithm with fitting violation function for analog neural array. In: **Neural Networks, 1996., IEEE International Conference on**. [S.l.: s.n.], 1996. v. 1, p. 565–570. Citations on pages [4](#) and [50](#).

BABU, A. R.; BABU, N. R. A generic approach for nesting of 2-d parts in 2-d sheets using genetic and heuristic algorithms. **Computer-Aided Design**, v. 33, p. 879 – 891, 2001. Citations on pages [4](#), [38](#), and [50](#).

BALDACCI, R.; BOSCHETTI, M. A.; GANOVELLI, M.; MANIEZZO, V. Algorithms for nesting with defects. **Discrete Applied Mathematics**, v. 163, Part 1, p. 17 – 33, 2014. *Matheuristics 2010*. Citations on pages [4](#) and [38](#).

BEASLEY, J. E. An exact two-dimensional non-guillotine cutting tree search procedure. **Operations Research**, v. 33, p. 49–64, 1985. Citations on pages [xv](#), [38](#), [39](#), [40](#), [53](#), and [55](#).

BENNELL, J. A.; DOWSLAND, K. A. Hybridising tabu search with optimisation techniques for irregular stock cutting. **Management Science**, v. 47, p. 1160–1172, 2001. Citations on pages [2](#) and [10](#).

BENNELL, J. A.; OLIVEIRA, J. F. The geometry of nesting problems: A tutorial. **European Journal of Operational Research**, v. 184, p. 397 – 415, 2008. Citations on pages [1](#), [9](#), and [31](#).

_____. A tutorial in irregular shape packing problems. **Journal of the Operational Research Society**, v. 60, n. S1, p. S93–S105, 2009. Citations on pages [1](#) and [10](#).

BENNELL, J. A.; SONG, X. A beam search implementation for the irregular shape packing problem. **Journal of Heuristics**, Springer US, v. 16, p. 167–188, 2010. Citations on pages [2](#) and [10](#).

BŁAŻEWICZ, J.; HAWRYLUK, P.; WALKOWIAK, R. Using a tabu search approach for solving the two-dimensional irregular cutting problem. **Annals of Operations Research**, Baltzer Science Publishers, Baarn/Kluwer Academic Publishers, v. 41, p. 313–325, 1993. Citations on pages [21](#) and [50](#).

BURKE, E.; HELLIER, R.; KENDALL, G.; WHITWELL, G. A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. **Operations Research**, v. 54, p. 587–601, 2006. Citations on pages [1](#) and [10](#).

BURKE, E. K.; HELLIER, R. S. R.; KENDALL, G.; WHITWELL, G. Irregular packing using the line and arc no-fit polygon. **Operations Research**, v. 58, p. 948–970, 2010. Citations on pages [1](#) and [10](#).

CARRAVILLA, M. A.; RIBEIRO, C.; OLIVEIRA, J. F. Solving nesting problems with non-convex polygons by constraint logic programming. **International Transactions in Operational Research**, Blackwell Publishing Ltd, v. 10, p. 651–663, 2003. Citations on pages [2](#) and [10](#).

CHERRI, L. H.; CARRAVILLA, M. A.; RIBEIRO, C.; TOLEDO, F. M. B. Optimality in nesting problems: New constraint programming models and a new global constraint for non-overlap. **Operations Research Perspectives**, v. 6, p. 100–125, 2019. Citation on page [2](#).

CHERRI, L. H.; MUNDIM, L. R.; ANDRETTA, M.; TOLEDO, F. M. B.; OLIVEIRA, J. F.; CARRAVILLA, M. A. Mixed-integer programming models for nesting problems. **European Journal of Operational Research**, v. 253, n. 3, p. 570–583, 2016. Citations on pages [3](#), [5](#), [30](#), [31](#), [34](#), [35](#), and [55](#).

DANIELS, K.; LI, Z.; MILENKOVIC, V. J. **Multiple Containment Methods**. [S.l.], 1994. 1-22 p. Citations on pages [2](#) and [10](#).

DOWSLAND, K. A.; DOWSLAND, W. B. Solution approaches to irregular nesting problems. **European Journal of Operational Research**, v. 84, n. 3, p. 506 – 521, 1995. Cutting and Packing. Citations on pages [1](#) and [10](#).

DOWSLAND, K. A.; DOWSLAND, W. B.; BENNELL, J. A. Jostling for position: Local improvement for irregular cutting patterns. **The Journal of the Operational Research Society**, v. 49, n. 6, p. 647–658, 1998. Citation on page [21](#).

DOWSLAND, K. A.; VAID, S.; DOWSLAND, W. B. An algorithm for polygon placement using a bottom-left strategy. **European Journal of Operational Research**, v. 141, p. 371 – 381, 2002. Citations on pages [1](#), [10](#), and [21](#).

EGEBLAD, J.; NIELSEN, B. K.; ODGAARD, A. Fast neighborhood search for two- and three-dimensional nesting problems. **European Journal of Operational Research**, v. 183, p. 1249 – 1266, 2007. Citations on pages [2](#) and [10](#).

ELKERAN, A. A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. **European Journal of Operational Research**, v. 231, p. 757 – 769, 2013. Citations on pages [2](#), [10](#), and [29](#).

ERDŐS, P.; GOODMAN, A. W.; PÓSA, L. The representation of a graph by set intersections. **Canadian Journal of Mathematics**, v. 18, p. 106–112, 1966. Citation on page [18](#).

FISCHER, A.; DAGLI, C. Employing subgroup evolution for irregular-shape nesting. **Journal of Intelligent Manufacturing**, Kluwer Academic Publishers, v. 15, p. 187–199, 2004. Citations on pages [4](#) and [38](#).

FISCHETTI, M.; LUZZI, I. Mixed-integer programming models for nesting problems. **Journal of Heuristics**, Kluwer Academic Publishers, v. 15, p. 201–226, 2009. Citations on pages [2](#), [3](#), [10](#), [29](#), and [30](#).

FOWLER, R. J.; PATERSON, M. S.; TANIMOTO, S. L. Optimal packing and covering in the plane are NP-complete. **Information Processing Letters**, v. 12, n. 3, p. 133 – 137, 1981. Citations on pages 1, 4, and 9.

FUJITA, K.; AKAGI, S.; HIROKAWA, N. Hybrid approach for optimal nesting using a genetic algorithm and a local minimization algorithm. In: **Proceedings of the 19th Annual ASME Design Automation Conference**. [S.l.: s.n.], 1993. v. 1, p. 477–484. Citations on pages 21 and 50.

FUJIYOSHI, K.; MURATA, H. Arbitrary convex and concave rectilinear block packing using sequence-pair. **Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on**, v. 19, n. 2, p. 224–233, Feb 2000. Citation on page 50.

GALINIER, P.; HERTZ, A. A survey of local search methods for graph coloring. **Computers & Operations Research**, v. 33, n. 9, p. 2547 – 2562, 2006. Citation on page 18.

GOLOMB, S. W. Checker boards and polyominoes. **The American Mathematical Monthly**, Mathematical Association of America, v. 61, p. 675–682, 1954. Citations on pages 3 and 37.

GOMES, A. M.; OLIVEIRA, J. F. Solving irregular strip packing problems by hybridising simulated annealing and linear programming. **European Journal of Operational Research**, v. 171, p. 811 – 829, 2006. Citations on pages 2, 3, 10, and 34.

GWEE, B.; LIM, M. Polyominoes tiling by a genetic algorithm. **Computational Optimization and Applications**, Kluwer Academic Publishers, v. 6, p. 273–291, 1996. Citations on pages 4 and 50.

HOPPER, E. **Two-dimensional packing utilising evolutionary algorithms and other meta-heuristic methods**. Phd Thesis (PhD Thesis) — University of Wales. Cardiff, 2000. Citation on page 21.

HU, Y.; HASHIMOTO, H.; IMAHORI, S.; YAGIURA, M. Efficient implementations of construction heuristics for the rectilinear block packing problem. **Computers & Operations Research**, v. 53, p. 206 – 222, 2015. Citation on page 50.

IMAMICHI, T.; YAGIURA, M.; NAGAMOCHI, H. An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem. **Discrete Optimization**, v. 6, p. 345 – 361, 2009. Citations on pages 2 and 10.

JAIN, S.; GEA, H. Two-dimensional packing problems using genetic algorithms. **Engineering with Computers**, Springer-Verlag, v. 14, p. 206–213, 1998. Citation on page 50.

JAKOBS, S. On genetic algorithms for the packing of polygons. **European Journal of Operational Research**, v. 88, p. 165 – 181, 1996. Citations on pages 21 and 50.

JONES, D. R. A fully general, exact algorithm for nesting irregular shapes. **Journal of Global Optimization**, v. 59, n. 2, p. 367–404, 2014. Citation on page 35.

KARP, R. Reducibility among combinatorial problems. In: MILLER, R.; THATCHER, J.; BOHLINGER, J. (Ed.). **Complexity of Computer Computations**. New York: Springer US, 1972, (The IBM Research Symposia Series). p. 85–103. Citation on page 18.

KASHKOUSH, M. N.; SHALABY, M. A.; ABDELHAFIEZ, E. A. A mixed-integer model for two-dimensional polyominoes strip packing and tiling problems. **International Journal of Operational Research**, Inderscience Publishers, v. 15, p. 391–405, 2012. Citations on pages [xv](#), [1](#), [5](#), [37](#), [38](#), [40](#), [41](#), [42](#), [48](#), [49](#), [51](#), [53](#), and [55](#).

KOU, L. T.; STOCKMEYER, L. J.; WONG, C. K. Covering edges by cliques with regard to keyword conflicts and intersection graphs. **Communications of the ACM**, v. 21, n. 2, p. 135–139, 1978. Citation on page [18](#).

LEAO, A. A.; TOLEDO, F. M.; OLIVEIRA, J. F.; CARRAVILLA, M. A. A semi-continuous mip model for the irregular strip packing problem. **International Journal of Production Research**, v. 54, n. 3, p. 712–721, 2016. Citations on pages [3](#) and [30](#).

LEAO, A. A.; TOLEDO, F. M.; OLIVEIRA, J. F.; CARRAVILLA, M. A.; ALVAREZ-VALDÉS, R. Irregular packing problems: A review of mathematical models. **European Journal of Operational Research**, v. 282, n. 3, p. 803 – 822, 2020. Citation on page [2](#).

LEIGHTON, F. T. A graph coloring algorithm for large scheduling problems. **Journal of Research of the National Bureau of Standards**, v. 84, n. 6, p. 489–506, 1979. Citations on pages [18](#) and [23](#).

LEUNG, S. C.; LIN, Y.; ZHANG, D. Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem. **Computers & Operations Research**, v. 39, p. 678 – 686, 2012. Citations on pages [2](#) and [10](#).

LI, Z. **Compaction algorithms for non-convex polygons and their applications**. Phd Thesis (PhD Thesis) — Harvard University, Cambridge, MA, 1994. Citations on pages [2](#) and [10](#).

LIN, J.-M.; CHEN, H.-L.; CHANG, Y.-W. Arbitrarily shaped rectilinear module placement using the transitive closure graph representation. **Very Large Scale Integration (VLSI) Systems, IEEE Transactions on**, v. 10, p. 886–901, Dec 2002. Citation on page [50](#).

LODI, A.; MARTELLO, S.; MONACI, M. Two-dimensional packing problems: A survey. **European Journal of Operational Research**, v. 141, n. 2, p. 241 – 252, 2002. Citations on pages [4](#) and [38](#).

MALAGUTI, E.; TOTH, P. A survey on vertex coloring problems. **International Transactions in Operational Research**, v. 17, n. 1, p. 1–34, 2010. Citation on page [18](#).

MANABE, S.; ASAI, H. A neuro-based optimization algorithm for tiling problems with rotation. In: **Neural Networks, 1999. IJCNN '99. International Joint Conference on**. [S.l.: s.n.], 1999. v. 6, p. 3750–3753 vol.6. Citations on pages [4](#) and [50](#).

MATAYOSHI, M. Two dimensional rectilinear polygon packing using genetic algorithm with a hierarchical chromosome. In: **Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on**. [S.l.: s.n.], 2011. p. 989–996. Citation on page [50](#).

MERLINI, D.; SPRUGNOLI, R.; VERRI, M. Strip tiling and regular grammars. **Theoretical Computer Science**, v. 242, p. 109 – 124, 2000. Citation on page [4](#).

MERLINI, D.; SPRUGNOLI, R.; VERRI, M. C. A strip-like tiling algorithm. **Theoretical Computer Science**, v. 282, p. 337 – 352, 2002. Citation on page [4](#).

MONSON, S. D.; PULLMAN, N. J.; REES, R. A survey of clique and biclique coverings and factorizations of $(0, 1)$ -matrices. **Bulletin of the ICA**, v. 14, p. 17–86, 1995. Citation on page 18.

MUNDIM, L. R.; ANDRETTA, M.; QUEIROZ, T. A. de. A biased random key genetic algorithm for open dimension nesting problems using no-fit raster. **Expert Systems with Applications**, v. 81, p. 358 – 371, 2017. Citation on page 2.

NIELSEN, B. K.; ODGAARD, A. **Fast neighbourhood search for nesting problem**. [S.l.], 2003. Citations on pages 1, 9, and 29.

OLIVEIRA, J.; FERREIRA, J. Algorithms for nesting problems. In: VIDAL, R. (Ed.). **Applied Simulated Annealing**. [S.l.]: Springer Berlin Heidelberg, 1993, (Lecture Notes in Economics and Mathematical Systems, v. 396). p. 255–273. Citations on pages 21 and 50.

OLIVEIRA, J. F.; GOMES, A. M.; FERREIRA, J. S. Topos - a new constructive algorithm for nesting problems. **OR-Spektrum**, Springer-Verlag, v. 22, p. 263–284, 2000. Citations on pages 1 and 10.

ORLIN, J. Contentment in graph theory: Covering graphs with cliques. **Indagationes Mathematicae (Proceedings)**, v. 80, n. 5, p. 406 – 424, 1977. Citation on page 18.

PARDALOS, P. M.; MAVRIDOU, T.; XUE, J. The graph coloring problem: A bibliographic survey. In: DU, D.-Z.; PARDALOS, P. (Ed.). **Handbook of Combinatorial Optimization**. Boston: Kluwer Academic Publishers, 1998. v. 2, p. 331–395. Citation on page 18.

PARGAS, R.; JAIN, R. A parallel stochastic optimization algorithm for solving 2d bin packing problems. In: **Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on**. [S.l.: s.n.], 1993. p. 18–25. Citations on pages 4 and 38.

PINHEIRO, P. R.; AMARO JR, B.; SARAIVA, R. D. A random-key genetic algorithm for solving the nesting problem. **International Journal of Computer Integrated Manufacturing**, (to appear), 2015. Citations on pages 2 and 10.

PULLMAN, N. J. Clique coverings of graphs — a survey. In: CASSE, L. R. A. (Ed.). **Combinatorial Mathematics X**. Berlin: Springer Berlin Heidelberg, 1983, (Lecture Notes in Mathematics, v. 1036). p. 72–85. Citation on page 18.

RATANAPAN, K.; DAGLI, C. H. An object-based evolutionary algorithm for solving rectangular piece nesting problems. In: **Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on**. [S.l.: s.n.], 1997. v. 2, p. 989–994. Citation on page 21.

REBENNACK, S.; REINELT, G.; PARDALOS, P. M. A tutorial on branch and cut algorithms for the maximum stable set problem. **International Transactions in Operational Research**, v. 19, n. 1-2, p. 161–199, 2012. Citations on pages 28 and 55.

RIBEIRO, C.; CARRAVILLA, M. A.; OLIVEIRA, J. F. Applying constraint logic programming to the resolution of nesting problems. **Pesquisa Operacional**, v. 19, p. 239–247, 1999. Citations on pages 2, 10, 21, and 35.

ROBERTS, F. S. Applications of edge coverings by cliques. **Discrete Applied Mathematics**, v. 10, n. 1, p. 93 – 109, 1985. Citation on page 18.

RODRIGUES, M. O. **Modelos matemáticos para o problema de empacotamento em faixas de peças irregulares**. Master's Thesis (Master's Thesis) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2015. Citation on page 47.

RODRIGUES, M. O.; CHERRI, L. H.; MUNDIM, L. R. Mip models for the irregular strip packing problem: new symmetry breaking constraints. **ITM Web Conf.**, v. 14, p. 00005, 2017. Citations on pages 6 and 29.

RODRIGUES, M. O.; MIYAZAWA, F. K. Fast constructive and improvement heuristics for edge clique covering. **Submitted to Journal of Heuristics**, 2018. Citations on pages 18 and 23.

RODRIGUES, M. O.; TOLEDO, F. M. B. A clique covering mip model for the irregular strip packing problem. **Computers & Operations Research**, v. 87, p. 221 – 234, 2017. Citations on pages 6 and 9.

SANTORO, M. C.; LEMOS, F. K. Irregular packing: Milp model based on a polygonal enclosure. **Annals of Operations Research**, v. 235, n. 1, p. 693–707, 2015. Citations on pages 3, 30, and 35.

SATO, A. K.; MARTINS, T. C.; TSUZUKI, M. S. G. An algorithm for the strip packing problem using collision free region and exact fitting placement. **Computer-Aided Design**, v. 44, p. 766 – 777, 2012. Citations on pages 2 and 10.

SHALABY, M.; KASHKOUSH, M. A particle swarm optimization algorithm for a 2-d irregular strip packing problem. **American Journal of Operations Research**, v. 3, p. 268–178, 2013. Citations on pages 4 and 38.

SIASOS, A.; VOSNIAKOS, G.-C. Optimal directional nesting of planar profiles on fabric bands for composites manufacturing. **{CIRP} Journal of Manufacturing Science and Technology**, v. 7, n. 3, p. 283 – 297, 2014. Citations on pages 4, 38, and 50.

TAKEFUJI, Y.; LEE, K.-C. A parallel algorithm for tiling problems. **Neural Networks, IEEE Transactions on**, v. 1, p. 143–145, Mar 1990. Citations on pages 4 and 50.

TOLEDO, F. M.; CARRAVILLA, M. A.; RIBEIRO, C.; OLIVEIRA, J. F.; GOMES, A. M. The dotted-board model: A new MIP model for nesting irregular shapes. **International Journal of Production Economics**, v. 145, p. 478 – 487, 2013. Citations on pages 3, 5, 10, 11, 12, 14, 15, 28, 30, and 55.

UMETANI, S.; YAGIURA, M.; IMAHORI, S.; IMAMICHI, T.; NONOBE, K.; IBARAKI, T. Solving the irregular strip packing problem via guided local search for overlap minimization. **International Transactions in Operational Research**, Blackwell Publishing Ltd, v. 16, p. 661–683, 2009. Citations on pages 2 and 10.

WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European Journal of Operational Research**, v. 183, p. 1109 – 1130, 2007. Citations on pages 1, 3, 4, 9, and 38.

WONG, W.; WANG, X.; MOK, P.; LEUNG, S.; KWONG, C. Solving the two-dimensional irregular objects allocation problems by using a two-stage packing approach. **Expert Systems with Applications**, v. 36, p. 3489 – 3496, 2009. Citations on pages 4 and 38.

WU, G.-M.; CHANG, Y.-C.; CHANG, Y.-W. Rectilinear block placement using b*-trees. In: **Computer Design, 2000. Proceedings. 2000 International Conference on**. [S.l.: s.n.], 2000. p. 351–356. Citation on page [50](#).

ZHANG, Y.; YANG, C. A generic approach for leather nesting. In: **Natural Computation, 2009. ICNC '09. Fifth International Conference on**. [S.l.: s.n.], 2009. v. 5, p. 303–307. Citations on pages [4](#) and [38](#).

SOLUTIONS - A CLIQUE COVERING MIP MODEL FOR THE IRREGULAR STRIP PACKING PROBLEM

Symbol * describes optimal solutions.

SHAPES15

blasz2*

BLAZEWICZ1*

BLAZEWICZ2*

BLAZEWICZ3*

BLAZEWICZ4*

BLAZEWICZ5

RCO1*

RCO2*

RCO3*

RCO4

RCO5

artif1_2*

artif2_4*

artif3_6

artif4_8

artif5_10

artif6_12

artif7_14

artif

shirts1_2*

shirts2_4*

shirts3_6*

shirts4_8

shirts5_10

fu5*

fu6*

fu7*

fu8*

fu9*

fu10*

fu

J1-10-10-0*

J1-10-10-1*

J1-10-10-2*

J1-10-10-3*

J1-10-10-4*

J1-12-20-0*

poly1a

poly1b

poly1c*

poly1d

poly1e

jakobs1*

SOLUTIONS - MIP MODELS FOR THE IRREGULAR STRIP PACKING PROBLEM: NEW SYMMETRY BREAKING CONSTRAINTS

Symbol * describes optimal solutions.

rcol1*

santoro1*

santoro2*

santoro3*

santoro4*

SOLUTIONS - INTEGER PROGRAMMING MODELS FOR THE QUASI-POLYOMINO STRIP PACKING PROBLEM

Symbol * describes optimal solutions.

Takefuji* 7 9

Asai1* 8 10

Asai2* 8 10

Asai3* 9 11

Asai4* 10 13

Matayoshi1-4* 20 23

Matayoshi1-5* 20 23

Matayoshi2-1* 22

Matayoshi2-2* 22

Matayoshi2-3* 22

Matayoshi2-4* 22

Matayoshi2-5* 22 28

0/Babu 4

Siasos* 7

Jain1* 10 12

Jain2* 10 11

Jain3*

20 22

Jain4*

11 12

Jain5*

15

Jain6

45

Fujiyoshi1*

9 10

Fujiyoshi2*

16 18

Lin1*

8 9

Lin2* 12 13

RL10* 11

RLT30 33 36

HHIY10_10* 10 11

HHIY10_100 32 34

HHIY100_100

401

T144_001*

12 13

T144_002*

1718

T144_004*

23 25

0/T144_008*

3234

T144_016*

45 48

T144_032

Blasz1_poly

5961

Blasz2.poly

Jakobs1_poly 123

Jakobs2.poly

289