A VNS Based Heuristic for a 2D Open Dimension Problem

Layane Rodrigues de Souza Queiroz and Thiago Alves de Queiroz(✉)

Institute of Mathematics and Technology, Federal University of Catalão, Catalão-Go 75704-020, Brazil
{layanequeiroz,taq}@ufcat.edu.br

Abstract. This paper is related to open-dimension problems in the area of cutting and packing. The problem we are interested in considers a set of irregularly shaped items and a two-dimensional (2D) bin in which one side is open. The objective is to pack all items in the bin and, in the case of a bin with one opened side, we also want to minimize the length of such a side. A packing cannot have items overlapping each other and items extrapolating the bin's dimensions. This problem appears in the metal-mechanic, textile, leather, and other related industries to the cutting of irregular pieces. We propose a variable neighborhood search-based heuristic for such a problem. A solution is coded as a vector of items that gives the sequence in which items will be packed. Neighborhood structures based on swap and insertion movements are considered in the local search phase, while the shaking phase contains a single neighborhood structure based on swap movements. Numerical experiments on benchmark instances show that the heuristic is competitive compared to other literature methods, obtaining equal or better solutions for 90.90% of the instances.

Keywords: Irregular cutting problems · Open dimension problems · Variable neighborhood search

1 Introduction

The problem of packing small objects inside one or more large objects appears in many real-world applications. In the logistics area, for example, we have pallets and boxes to be loaded into containers, trucks, trains, ships, or airplanes. This problem is computationally hard in many of its variants and so mathematical programming models and heuristics have been proposed in the literature. It is important to mention that, from a theoretical point of view, it is equivalent to cutting problems, which in turn requires the cutting of large objects to produce small ones. In the textile and metal-mechanic industry, for example, fabric rolls and metal plates are cut to produce pieces of products. For an overview of packing and cutting problems, we refer to the book in [20].

In this paper, we are interested in problems where small objects can have irregular shapes. We look for the packing (cutting) of all small objects (hereafter

A. Sleptchenko et al. (Eds.): ICVNS 2022, LNCS 13863, pp. 125–136, 2023.
https://doi.org/10.1007/978-3-031-34500-5_10

called items) in a single large object (hereafter called a strip). This problem is known as the two-dimensional irregular strip packing problem [18]. As the strip is assumed to have a fixed width, the objective is to minimize its opened length by obtaining a feasible packing. Concerning irregularly shaped items, the guarantee of a feasible packing can be obtained with geometric tools, such as the raster method, the phi-functions, the direct trigonometry, and the no-fit polygons. For these tools, we refer to the tutorial in [3]. We use the no-fit raster, a combination of the raster method and the no-fit polygons [23].

In the literature on the two-dimensional irregular strip packing problem, we may find different contributions, from simple mathematical programming models to sophisticated heuristics. As this problem is NP-hard, most of the contributions are related to heuristics. In [1], sequences of items are packed with the bottom-left rule. In this rule, an item is translated to the bottom and then to the left in the strip. This rule is also used in [15], where a genetic algorithm generates the sequence of items; in [10], where a 2-exchange heuristic generates the sequence of items; in [11], where simulated annealing is used for generating the sequence of items; in [17], where a biased random key genetic algorithm generates the sequence of items.

Other contributions are related to a constraint programming model in [4], the integration of the cuckoo search with a guided local search in [7], and a tailored branch-and-cut algorithm, where a variable neighborhood search heuristic generates feasible solutions, in [22]. Integer linear programming models are proposed in [5,8,16,19,23]. The model in [19] uses clique constraints to detect infeasible packings. They improved most of the previous solutions presented in the literature.

We propose a Variable Neighborhood Search (VNS) for the two-dimensional irregular strip packing problem. The VNS's neighborhood structures are based on swap and insertion movements. The shaking phase consists of a single structure based on swap movements, while the local search consists of the variable neighborhood descent (VND). The VNS generates the sequence of items, while a function is used to transform the given sequence into a feasible packing. For that, items are positioned in the strip by combining the bottom-left and top-left placement rules. Results obtained with the VNS are compared with those in [19,22], with better solutions for 27.27% of the instances and equal solutions for 63.63% of the instances.

The remainder of this paper is organized as follows. In Sect. 2, we define the problem and the geometric tools used to guarantee feasible packings. In Sect. 3, we present the variable neighborhood search and how a sequence of items is transformed into a problem solution. In Sect. 4, we perform computational experiments on literature instances and compare the performance of the VNS with the literature. In Sect. 5, we give some conclusions and directions for future works.

2 Problem Definition

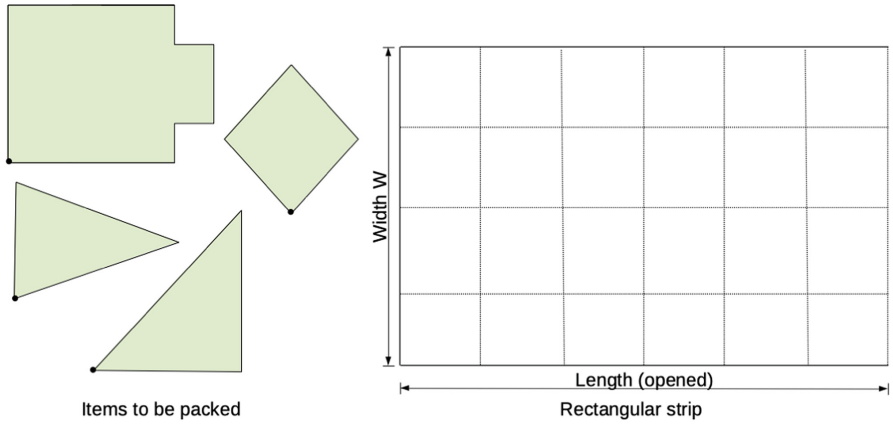
This paper is about the Two-Dimensional Irregular Strip Packing Problem (2ISP). We assume the strip is rectangular, while items are defined as (irregular) polygons without holes. Each item j has a set of vertices V_j, an area a_j, and a reference vertex p_j. We assume an item is positioned in the strip by its reference vertex, which in turn is defined as the vertex with the lowest y-coordinate and, in the case of ties, with the lowest x-coordinate. A solution is built on the Cartesian plane. The strip's lower-left coordinates are at $(0, 0)$ and its top-right coordinates are at (∞, W). We associate the opened length (∞) to the x-axis and the width W to the y-axis. The problem's objective is to minimize the opened length while packing all items in the strip.

We assume the strip is discrete and then defined by a grid of points [2]. The reference vertex of items is positioned on points of this grid. A feasible solution (packing) is obtained when all items are packed inside the strip (i.e., there is no part/area of any item extrapolating the strip's dimensions) and items do not overlap each other (i.e., there is no intersection between any two items when positioned on the grid). To guarantee these two conditions to obtain a feasible solution, we calculate the inner-fit raster of each item with the strip and the no-fit raster between any two items. Figure 1 shows an example of irregularly shaped items, a rectangular strip, and the no-fit raster between two items.

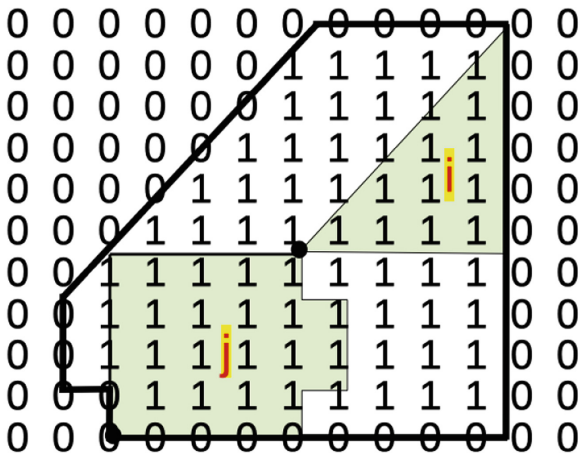
The inner-fit and no-fit rasters are calculated in a pre-processing step [23]. For the inner-fit raster, each item j is positioned by its reference vertex at the lowest-left position on the grid, touching the strip's borders where possible but not extrapolating the strip's dimensions. Then, this item is translated around the strip always touching the strip's borders. The inner-fit polygon that is generated is next discretized according to the strip's grid. Positions having value "1" mean that such an item cannot be positioned there since it does not respect the strip's dimensions. For the no-fit raster, we consider each pair of items i and j. Item i is fixed on the plane, while j is positioned in such a way that it touches i. Then, item j is translated (by its reference vertex) around and touching i. The no-fit polygon that is generated is next discretized according to the strip's grid. Positions having value "1" mean that item j cannot be positioned there since such items will overlap each other.

3 Proposed Heuristic

We develop a VNS heuristic to the 2ISP. This heuristic has been applied to solve many continuous and discrete optimization problems, obtaining very competitive results compared to other literature methods [6,13]. Differently from other literature contributions that applied VNSs to irregular cutting problems, we consider the shaking phase defined on only one neighborhood structure, while the local search phase is composed of three neighborhood structures. The idea is to prioritize the local search phase to obtain high-quality solutions. As this phase could require a large computational time, the VNS has the advantage of



(a) Example of items and strip.



(b) Example of no-fit raster between items i and j.

Fig. 1. Illustrative example for the 2ISP.

carrying the optimization over a single solution and thus helping on reducing the computational effort. Algorithm 1 presents the proposed VNS.

In Algorithm 1, we code the solution x as a vector of integers. This is commonly adopted in the literature on irregular cutting problems [21,22]. Each integer represents the index of an item in the input instance. This means that x contains the sequence in which items are packed in the strip's grid. In the shaking phase, we consider that only the neighborhood structure N_1 is applied, where positions i and j are randomly chosen. On the other hand, the local-search phase considers three neighborhood structures, which are N_1, N_2, and N_3. In detail, the neighborhood structures are:

- N_1 (one-element swap): The elements of two given positions i and j are swapped, i.e., $i \leftrightarrow j$;
- N_2 (one-element insertion): Given two positions i and j, position i is inserted immediately after position j;
- N_3 (three-elements change): The elements of three given positions i, j, and u are changed, i.e., $i \rightarrow j$, $j \rightarrow u$, and $u \rightarrow i$. Notice that auxiliary variables are used to avoid losing information.

Algorithm 1: VNS proposed to the 2ISP

```
1  x ← randomly generated solution
2  for a given number of iterations do
3      while true do
           /*shaking phase*/
4          x' ← random solution in the neighborhood structure N_1(x)
           /*local-search phase*/
5          k ← 1
6          while k ≤ 3 do
7              x'' ← first solution in the neighborhood structure N_k(x') that
               is better than x', if one exists
8              if F(x'') < F(x') then
9                  x' ← x''; k ← 1
10             else
11                 k ← k + 1
           /*change of neighborhood*/
12         if F(x') < F(x) then
13             x ← x'
14         else
15             break
```

The local-search phase in Algorithm 1 consists of the variable neighborhood descent heuristic [12]. It starts by looking for the first solution in the neighborhood structure $N_k(x')$, initially for $k = 1$, that is better than x', the solution of the shaking phase. If this is true, solution x' is updated and the search continues on the same neighborhood structure; otherwise, the search continues on the next neighborhood structure. It is worth mentioning that all possibilities of positions in N_1, N_2, and N_3 are tested until finding the first improved solution if one exists. After the local search, solution x' is compared with the current solution x. The latter is updated if x' is better; otherwise, the while loop is broken.

The value of a solution x is determined by the function $F()$ in Algorithm 2. This function is based on the decoder proposed by [22]. The difference is that we are using only one placement rule, which is a combination of the bottom-left and top-left rules. The proposed placement rule divides the solution vector x into two parts. The left half of vector x assumes that items are positioned by

the bottom-left rule, while the right half part has its items positioned by the top-left rules.

Algorithm 2: Cost of a solution x calculated by the function $F()$

```
if solution x is in the hash table then
    return length L of the packing defined by x
S ← ∅
foreach item j in solution x do
    if j is in the left half of x then
        (a, b) ← the first point of the grid by the bottom-left rule that
        gives a feasible packing for j
        S ← S ∪ {j, (a, b)}
    else
        (a, b) ← the first point of the grid by the top-left rule that gives a
        feasible packing for j
        S ← S ∪ {j, (a, b)}
Save x and S in the hash table
return length L of the packing S defined by x
```

Figure 2 has an example of Algorithm 2 applied to a solution with four items in the sequence $\{1, 4, 3, 2\}$. Items 1 and 4 are in the left half of x and then are packed by the bottom-left rule. Items 3 and 2 are in the right half part and then are packed by the top-left rule. The resulting packing has length L, which is the cost of the solution returned by the function $F()$.

4 Computational Experiments

We coded all algorithms in the C++ programming language and performed computational experiments on literature instances. The experiments are executed in a computer with an Apple M2 processor, 8 GB of RAM, and macOS 13 as the operating system. The proposed VNS has a single parameter to define, which is the maximum number of iterations. We define it as a maximum time limit, set to 120 s, to solve each instance. The VNS runs 5 times and the best solution found among these is reported.

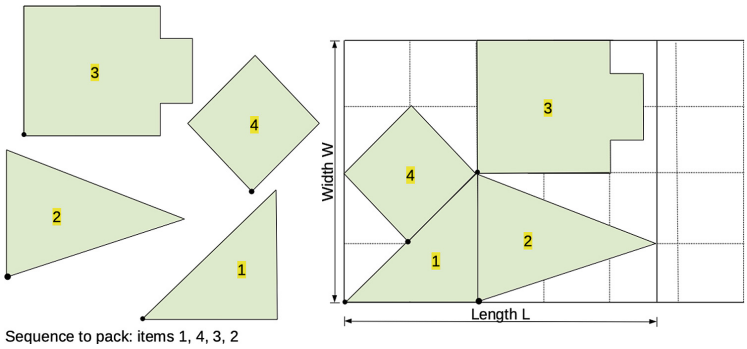


Fig. 2. Example of packing obtained with the function $F()$ for a solution x.

Table 1. Data of the 22 instances.

Instance name	Authors	Number of items	Strip's width (W)
blazewicz1	Toledo et al. [23]	7	15
blazewicz2	Toledo et al. [23]	14	15
blazewicz3	Toledo et al. [23]	21	15
blazewicz4	Toledo et al. [23]	28	15
blazewicz5	Toledo et al. [23]	35	15
dagli1	Rodrigues and Toledo [19]	10	60
fu	Fujita et al. [9]	12	38
poly1a	Hopper [14]	15	40
poly1b	Rodrigues and Toledo [19]	15	40
poly1c	Rodrigues and Toledo [19]	15	40
poly1d	Rodrigues and Toledo [19]	15	40
poly1e	Rodrigues and Toledo [19]	15	40
shapes2	Toledo et al. [23]	8	40
shapes4	Toledo et al. [23]	16	40
shapes5	Toledo et al. [23]	20	40
shapes7	Toledo et al. [23]	28	40
shapes15	Toledo et al. [23]	43	40
shirts1-2	Rodrigues and Toledo [19]	13	40
shirts2-4	Rodrigues and Toledo [19]	26	40
shirts3-6	Rodrigues and Toledo [19]	39	40
shirts4-8	Rodrigues and Toledo [19]	52	40
shirts5-10	Rodrigues and Toledo [19]	65	40

We consider 22 instances from the literature, which may be found on the website of the EURO Special Interest Group on Cutting and Packing[1]. Table 1

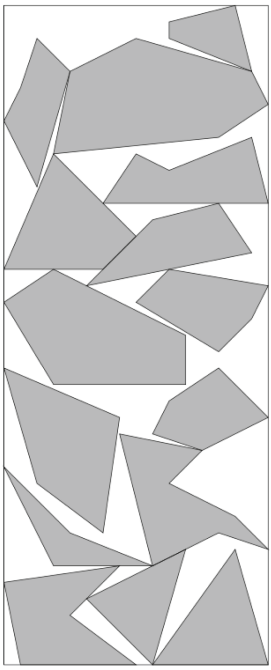
[1] https://www.euro-online.org/websites/esicup/data-sets.

has the instance name, the authors who proposed the instance, the total number of items, and the strip width. Concerning the grid of points, we discretize the strip, the inner-fit rasters, and no-fit polygons by one unit of distance according to [19].

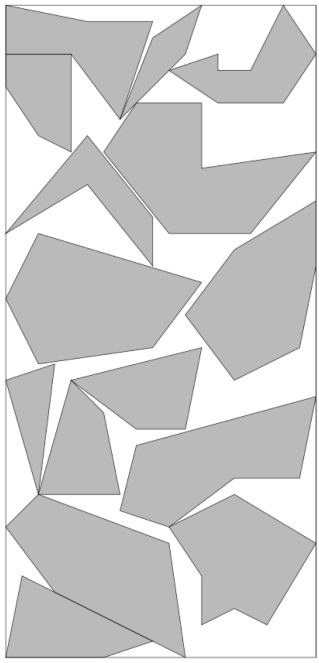
Table 2 has the results obtained with the proposed VNS and two other literature methods, i.e., the branch-and-cut algorithms in [19,22]. The algorithm in [19] is not able to obtain the solution of two instances, namely shapes15 and shirts5-10. On the other hand, the VNS and the algorithm in [22] report a solution to all instances. The proposed VNS obtains equal solutions for 14 out of 22 instances. For the others, the VNS improves the solution of 6 instances, namely poly1a, poly1b, shapes5, shapes7, shapes9, and shapes15. On the other hand, the VNS is worse for instances shirts2-4 and shirts3-6, differing from one unit in terms of length. The computing time of the VNS is not reported in the table because it is used as the stopping criterion and is equal to 120 s for each instance. In Fig. 4, we show the improved solutions obtained with the proposed VNS.

Table 2. Comparing the VNS with two other literature algorithms.

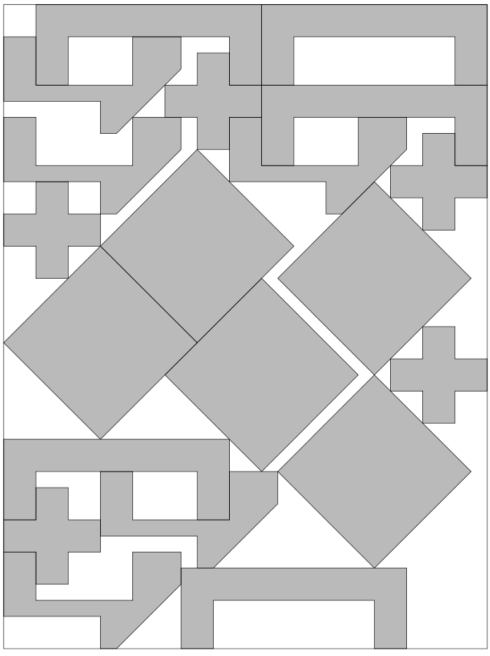
Instance	Proposed VNS	Souza Queiroz and Andretta [22]		Rodrigues and Toledo [19]	
	Length L	Length L	Time (s)	Length L	Time (s)
blazewicz1	8	8	7.43	8	0.01
blazewicz2	14	14	195.39	14	4.17
blazewicz3	21	21	3600.00	20	1139.96
blazewicz4	28	28	3600.00	27	3600.00
blazewicz5	35	35	3600.00	34	3600.00
dagli1	23	23	1.48	23	100.73
fu	34	34	3600.00	37	3600.00
poly1a	**16**	17	3600.00	17	3600.00
poly1b	**19**	20	3600.00	20	3600.00
poly1c	13	13	63.78	13	152.25
poly1d	13	13	3600.00	13	3600.00
poly1e	12	12	3600.00	12	3600.00
shapes2	14	14	2.10	14	1.09
shapes4	25	25	2431.50	25	3600.00
shapes5	**30**	31	3600.00	31	3600.00
shapes7	**41**	42	3600.00	45	3600.00
shapes9	**48**	49	3600.00	54	3600.00
shapes15	**61**	62	3600.00	–	3600.00
shirts1-2	13	13	0.03	13	0.02
shirts2-4	18	**17**	177.29	**17**	47.77
shirts3-6	25	**24**	3558.46	**24**	497.68
shirts4-8	33	33	3600.00	33	3600.00
shirts5-10	41	41	3600.00	–	3600.00



(a) poly1a

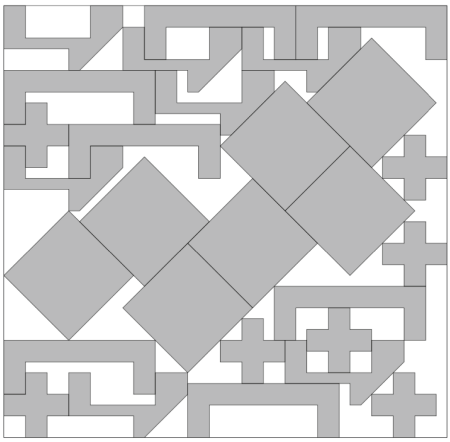


(b) poly1b

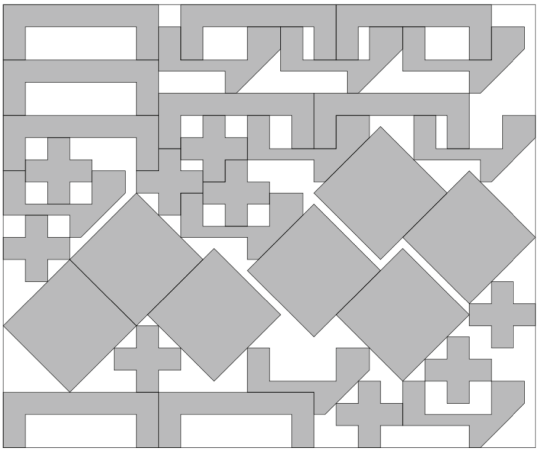


(c) shapes5

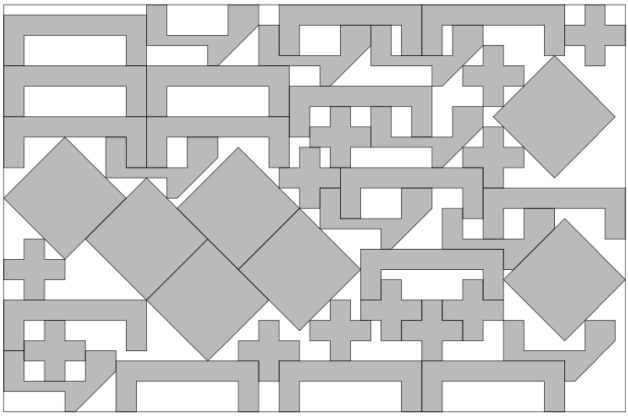
Fig. 3. Solutions improved by the proposed VNS - part 1.



(a) shapes7



(b) shapes9



(c) shapes15

Fig. 4. Solutions improved by the proposed VNS - part 2.

5 Concluding Remarks

This paper is related to the two-dimensional irregular strip packing, an open-dimension problem, for which the strip's length is minimized while packing all items. Besides being an NP-hard problem, it is found in many real-world applications. As the items to pack may have an irregular shape, we use geometric tools such as the inner-fit raster and no-fit raster to guarantee feasible solutions. The proposed VNS has the shaking phase defined over a single neighborhood structure, while the local search as the VND has three neighborhoods based on swap and insertion movements. Due to the solution representation, we define a function to obtain the packing and so its length. In this function, items in the given sequence are packed by a combination of the bottom-left and top-left placement rules (Fig. 2).

The computational experiments on literature instances show the proposed VNS is competitive, obtaining equal or better solutions for 90.90% of the instances. For the other instances, the difference is one unit in the strip's length, which is relatively small. We notice that there is room for improvement in many directions. One could be in the proposal of new ways to code and decode a solution, as in the case of defining new placement rules. Further exploration of the scale adopted to the grid could also be worthwhile to identify the trade-off between solution quality and computing time. Another interesting direction is the combination of heuristics and mathematical programming models. It could be important to have a comparison between different paradigms, e.g., single trajectory heuristics versus population-based ones. In terms of instances, one direction could be to have items with holes and allow the rotation of items.

Acknowledgements. The authors acknowledge the financial support of the National Council for Scientific and Technological Development (CNPq grants numbers 405369/2021-2 and 311185/2020-7) and the State of Goiás Research Foundation (FAPEG).

References

1. Albano, A., Sapuppo, G.: Optimal allocation of two-dimensional irregular shapes using heuristic search methods. IEEE Trans. Syst. Man Cybern. **10**(5), 242–248 (1980)
2. de Almeida Cunha, J.G., de Lima, V.L., de Queiroz, T.A.: Grids for cutting and packing problems: a study in the 2D knapsack problem. 4OR **18**(3), 293–339 (2019). https://doi.org/10.1007/s10288-019-00419-9
3. Bennell, J.A., Oliveira, J.F.: A tutorial in irregular shape packing problems. J. Oper. Res. Soc. **60**, 93–105 (2009)
4. Carravilla, M.A., Ribeiro, C., Oliveira, J.F., Gomes, A.M.: Solving nesting problems with non-convex polygons by constraint logic programming. Int. Trans. Oper. Res. **10**, 651–663 (2003)
5. Cherri, L.H., Mundim, L.R., Andretta, M., Toledo, F.M., Oliveira, J.F., Carravilla, M.A.: Robust mixed-integer linear programming models for the irregular strip packing problem. Eur. J. Oper. Res. **253**(3), 570–583 (2016)

6. Duarte, A., Pardo, E.G.: Special issue on recent innovations in variable neighborhood search. J. Heuristics **26**, 335–338 (2020)
7. Elkeran, A.: A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. Eur. J. Oper. Res. **231**(3), 757–769 (2013)
8. Fischetti, M., Luzzi, I.: Mixed-integer programming models for nesting problems. J. Heuristics **15**, 201–226 (2009)
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, pp. 477–484. Albuquerque, New Mexico, USA (1993)
10. Gomes, A.M., Oliveira, J.F.: A 2-exchange heuristic for nesting problems. Eur. J. Oper. Res. **141**(2), 359–370 (2002)
11. Gomes, A.M., Oliveira, J.F.: Solving irregular strip packing problems by hybridising simulated annealing and linear programming. Eur. J. Oper. Res. **171**(3), 811–829 (2006)
12. Hansen, P., Mladenović, N., Pérez, J.M.: Variable neighbourhood search: methods and applications. Ann. Oper. Res. **175**(1), 367–407 (2010)
13. Hansen, P., Mladenović, N., Todosijević, R., Hanafi, S.: Variable neighborhood search: basics and variants. EURO J. Comput. Optim. **5**, 423–454 (2017)
14. Hopper, E.: Mathematical models and heuristic methods for nesting problems. School of Engineering, University of Wales, Cardiff, Doutorado (2000)
15. Jakobs, S.: On genetic algorithms for the packing of polygons. Eur. J. Oper. Res. **88**(1), 165–181 (1996)
16. Leão, A.A.S., Toledo, F.M.B., Oliveira, J.F., Carravilla, M.A.: A semi-continuous mip model for the irregular strip packing problem. Int. J. Prod. Res. **54**(3), 712–721 (2016)
17. Mundim, L.R., Andretta, M., Queiroz, T.A.: A biased random key genetic algorithm for open dimension nesting problems using no-fit raster. Expert Syst. Appl. **81**, 358–371 (2017). https://doi.org/10.1016/j.eswa.2017.03.059
18. Oliveira, J.F.C., Ferreira, J.A.S.: Algorithms for nesting problems. In: Vidal, R.V.V. (eds.) Applied Simulated Annealing. Lecture Notes in Economics and Mathematical Systems, vol. 396, pp. 255–273. Springer, Berlin, Heidelberg (1993). https://doi.org/10.1007/978-3-642-46787-5_13
19. Rodrigues, M.O., Toledo, F.M.: A clique covering mip model for the irregular strip packing problem. Comput. Oper. Res. **87**, 221–234 (2017)
20. Scheithauer, G.: Introduction to Cutting and Packing Optimization: Problems, Modeling Approaches, Solution Methods, vol. 263. Springer, Cham (2017)
21. Souza, Queiroz, L.R., Andretta, M.: Two effective methods for the irregular knapsack problem. Appl. Soft Comput. **95**, 106485 (2020)
22. Souza, Queiroz, L.R., Andretta, M.: A branch-and-cut algorithm for the irregular strip packing problem with uncertain demands. Int. Trans. Oper. Res. **29**(6), 3486–3513 (2022)
23. Toledo, F.M.B., Carravilla, M.A., Ribeiro, C., Oliveira, J.F., Gomes, A.M.: The dotted-board model: a new mip model for nesting irregular shapes. Int. J. Prod. Econ. **145**(2), 478–487 (2013)