

Article

COMPACT: Concurrent or Ordered Matrix-Based Packing Arrangement Computation Technique

Gokhan Serhat 

Max Planck Institute for Intelligent Systems, 70569 Stuttgart, Germany; serhat@is.mpg.de

Abstract: Despite their versatility in treating irregular geometries, the raster methods have received limited attention in solving packing problems involving rotatable objects. In addition, raster approximation allows the use of unique performance metrics and indirect consideration of constraints, which have not been exploited in the literature. This study presents the Concurrent or Ordered Matrix-based Packing Arrangement Computation Technique (COMPACT). The method allows the objects to be rotated by arbitrary angles, unlike the right-angled rotation restrictions imposed in many existing packing optimization studies based on raster methods. The raster approximations are obtained through loop-free operations that improve efficiency. Additionally, a novel performance metric is introduced, which favors efficient filling of the available space by maximizing the overall contact within the domain. Moreover, the objective functions are exploited to discard the overlap and overflow constraints and enable the use of unconstrained optimization methods. The results of the case studies demonstrate the effectiveness of the proposed technique.

Keywords: packing; heuristics; raster representation; concurrent optimization; rotatable objects



Citation: Serhat, G. COMPACT: Concurrent or Ordered Matrix-Based Packing Arrangement Computation Technique. *Appl. Sci.* **2021**, *11*, 5217. <https://doi.org/10.3390/app11115217>

Academic Editor: Panagiotis G. Asteris

Received: 7 May 2021

Accepted: 1 June 2021

Published: 4 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Packing optimization is an important problem class that concerns various disciplines including warehouse management, shipping, and design of engineering products. The broad nature of this problem requires robust and efficient algorithms that are easy to utilize.

Various techniques have been developed to solve different packing optimization problems. A large amount of available studies have addressed the problem of packing two-dimensional rectangular objects into rectangular bins while keeping the object edges orthogonal to the domain boundaries. For instance, Chen and Huang [1] computed solutions for two-dimensional rectangle packing problems by introducing a search algorithm that uses a corner occupation strategy to individually place the objects into the container. Later, Leung et al. [2] proposed a hybrid heuristic algorithm that combines greedy strategy and simulated annealing methods to solve the rectangular knapsack packing problem. Shiangjen et al. [3] studied the same problem by proposing a bidirectional heuristic placement algorithm that attempts to fit rectangles around both left and right boundaries of the domain. Chen et al. [4] subsequently developed a deterministic heuristic algorithm for solving the two-dimensional rectangular packing problems that require maximizing the filled portion of the rectangular domain. Recently, Li et al. [5] proposed an adaptive genetic algorithm for solving two-dimensional packing problems that aim to maximize the filling rate of a rectangular region.

Packing problems concerning nonrectangular objects and domains have also been prevalently investigated. For example, Del Valle et al. [6] developed heuristic approaches for solving two-dimensional cutting/packing problems involving irregular objects. Later, Chen et al. [7] proposed a heuristic algorithm for solving circle packing problems for placing multiple identical circles into the smallest circular container. Lately, Bouzid and Salhi [8] investigated orthogonal packing of rectangular objects into a circular space of fixed radius.

The complexity of the packing problems notably increases when the nonorthogonal rotation of the objects is allowed in addition to the translation [9]. These problems received relatively lesser attention compared to the problems involving fixed object orientations or orthogonal rotations. In one study, Zhang and Zhang [10] introduced a two-dimensional packing optimization technique called the finite-circle method, which approximates polygons with a set of circles and converts the non-overlapping constraints into simple constraints between circles. Martins and Tsuzuki [11] used a heuristic approach to minimize the waste space during the rotational placement of two-dimensional polygons into irregular containers. In another study by Martinez-Sykora et al. [12], a strategy was proposed to solve the bin packing problems with irregular pieces that can be freely rotated.

The difficulties in the formulations for packing problems involving irregular objects have led to the development of raster point methods, which rely on approximating the objects on a discretized grid. With these methods, packing configurations can be represented by assigning zeroes to the free domain cells and different values to the cells where a single object exists or multiple objects overlap. The key advantage of raster techniques is their applicability to the problems involving objects with complex shapes, where the main drawback is their resolution-dependent precision [13]. In the literature, the research on packing optimization with raster methods is limited compared to the available work based on other techniques. In a particular study, Toledo et al. [14] proposed a discrete modeling approach called the Dotted-Board Model, which relies on representing the bin with a two-dimensional grid and using the plausible grid points for the placement of arbitrary polygons. Later, Mundim et al. [15] extended the Dotted-Board Model for packing non-polygonal objects with curved edges and used it within a heuristic approach based on genetic algorithms. Recently, Sato et al. [16] developed the raster overlap minimization algorithm to solve the irregular strip packing problem using a raster method that limits the solution space.

This study presents the new Concurrent or Ordered Matrix-based Packing Arrangement Computation Technique (COMPACT). In this framework, the object shapes are approximated by matrix-based (raster) representations. The objects can be rotated by angles that are not necessarily restricted to be right-angle multiples, unlike the conditions imposed in many existing packing optimization studies relying on raster methods. The raster approximations are obtained through loop-free operations that improve efficiency. In addition, a novel performance metric that aids efficient filling of the available space is introduced. This metric promotes maximizing the internal contact between the objects as well as the contact between the objects and domain boundaries. Moreover, the possibility of exploiting the objective functions for discarding the overlap and overflow constraints is explored to enable the use of unconstrained optimization methods. As case studies, the developed technique is utilized to pack multiple rectangular and circular objects into square domains using concurrent and ordered placement strategies. The results show that the proposed technique performs effectively in determining the packing arrangements. The organization of the study is as follows: The matrix-based object representation that forms the backbone of the developed technique is described in Section 2. Section 3 covers the details regarding the formulation of the optimization problem. Section 4 contains the results from various case studies. In Section 5, the conclusions are stated and possible directions for future studies are outlined.

2. Matrix-Based Object Representation

The proposed technique relies on the raster representation of the objects in the matrix space. Let a rectangular design space be described by an empty matrix with H rows of height and W columns of width. A rectangular object having a height of h rows and a width of w columns can now be modeled within the domain by filling an area consisting of $h \times w$ entities with ones. The geometries should be approximated for curved shapes and polygons whose edges are not aligned with the rows and columns. Figure 1 shows (a) exemplary matrix space approximations of a circle and a rectangle whose edges are

aligned with the horizontal and vertical domain axes, and (b) the sparsity pattern of the matrix containing the objects.

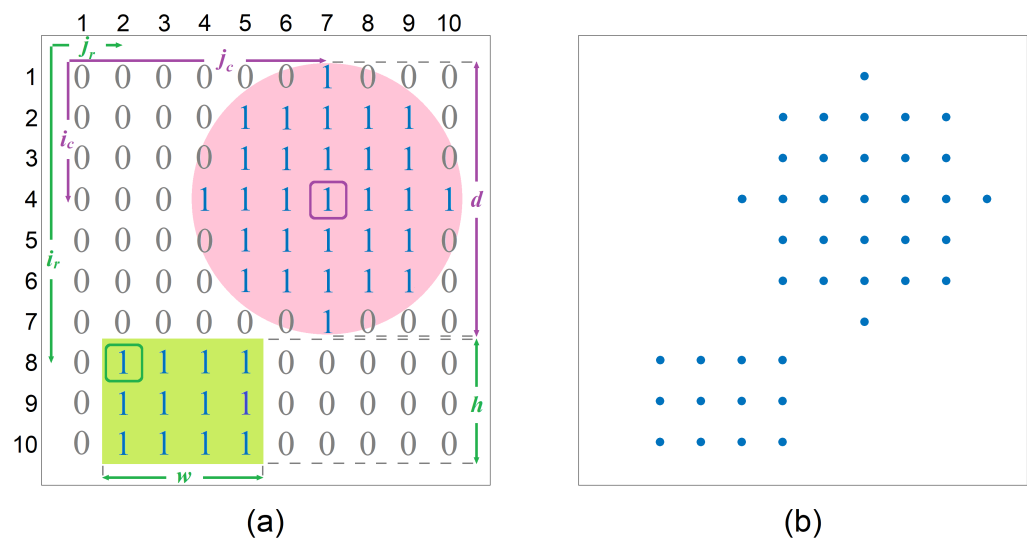


Figure 1. (a) Matrix space approximations of a rectangle with $h = 3$ rows height and $w = 4$ columns width, and a circle with $d = 7$ cells diameter. The upper-left corner of the rectangle and center of the circle are located on $(i_r = 8, j_r = 2)$ and $(i_c = 4, j_c = 7)$, respectively. (b) The sparsity pattern of the matrix containing the objects.

The proposed technique does not restrict the edges of the rectangular objects to be aligned with the domain boundaries. Raster approximation of the rotated objects is exemplified with a $h = 0.4H = 0.4W$ by $w = 1.5h$ rectangle rotated by an angle of $\theta = 30^\circ$ around its top-left corner. Figure 2 shows the sparsity patterns of this object for domain sizes of (a) 20×20 , (b) 50×50 , and (c) 100×100 , which results in 92, 595, and 2414 nonzero entities (N_{nz}), respectively. As seen in this figure, the shape of the rotated geometry is more accurately captured with increasing resolution. A mathematical convergence can also be observed by calculating the ratio of nonzero entities to the total number of domain entities. When an unrotated rectangle is considered, the exact filling ratio (the proportion of ones to the total number of elements) can be calculated as $hw/HW = 0.24$. For the presented three cases, the filling ratios are computed as $92/400 = 0.2300$, $595/2500 = 0.2380$, and $2414/10,000 = 0.2414$, which converge to the exact value with higher resolution.

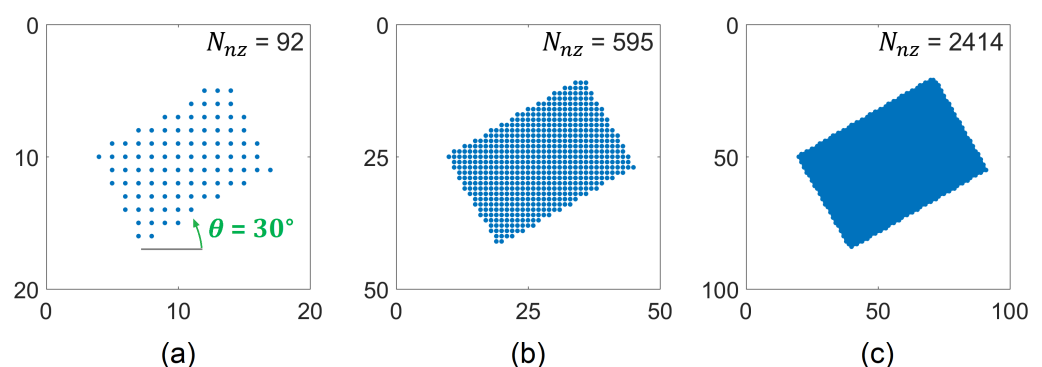


Figure 2. Sparsity patterns of a rectangle rotated by an angle of $\theta = 30^\circ$ around its top-left corner for domain sizes of (a) 20×20 , (b) 50×50 , and (c) 100×100 ; which result in 92, 595, and 2414 nonzero entities, respectively. The rectangles have a size of $(h = 0.4H, w = 1.5h)$.

Increasing the number of raster points inherently lengthens the computation durations [13]. Therefore, excessive discretization resolutions should be avoided to keep the computation durations within reasonable limits.

The raster approximations for the rectangles are attained by performing vector inequality operations and elementwise matrix multiplications. In the present study, only top-left corner coordinates of the rectangles are restricted to be integers. The other corners do not necessarily coincide with the matrix grid unlike the conditions enforced with certain raster-based methods such as the Dotted-Board Model [14,15].

For a rectangle rotated counterclockwise by θ degrees, the row and column coordinates of the bottom right corner (i_θ and j_θ) can be found as

$$\begin{aligned} i_\theta &= i_r + h \cos(\theta) - w \sin(\theta) \\ j_\theta &= j_r + h \sin(\theta) + w \cos(\theta) \end{aligned} \quad (1)$$

Let r_H be a row vector and c_W be a column vector with elements ranging from 1 to H and 1 to W , respectively. Then, the lower and upper boundaries for the rows (i_l and i_u) as well as the columns (j_l and j_u) can be computed as

$$\begin{aligned} i_l &= \lfloor i_r - \tan(\theta)(c_W - j_r) \rfloor \\ i_u &= \lfloor i_\theta - \tan(\theta)(c_W - j_\theta) \rfloor \\ j_l &= \lfloor j_r + \tan(\theta)(r_H - i_r) \rfloor \\ j_u &= \lfloor j_\theta + \tan(\theta)(r_H - i_\theta) \rfloor \end{aligned} \quad (2)$$

where the brackets ' $\lfloor \cdot \rfloor$ ' signify rounding towards the nearest integer. Note that, i_l and i_u are column vectors where j_l and j_u are row vectors. Using the calculated boundary vectors, four constituent matrices can be obtained through the following vector inequalities:

$$\begin{aligned} D_1 &= (r_H \geq i_l), & D_2 &= (r_H < i_u) \\ D_3 &= (c_W \geq j_l), & D_4 &= (c_W < j_u) \end{aligned} \quad (3)$$

The final object representation is attained by performing element-by-element multiplication (denoted by operator ' \circ ') of calculated constituent matrices:

$$D = D_1 \circ D_2 \circ D_3 \circ D_4 \quad (4)$$

The described approach avoids using loops, which in turn substantially accelerates the computations. Figure 3 demonstrates the constituent matrices and final representation for a sample rectangular object.

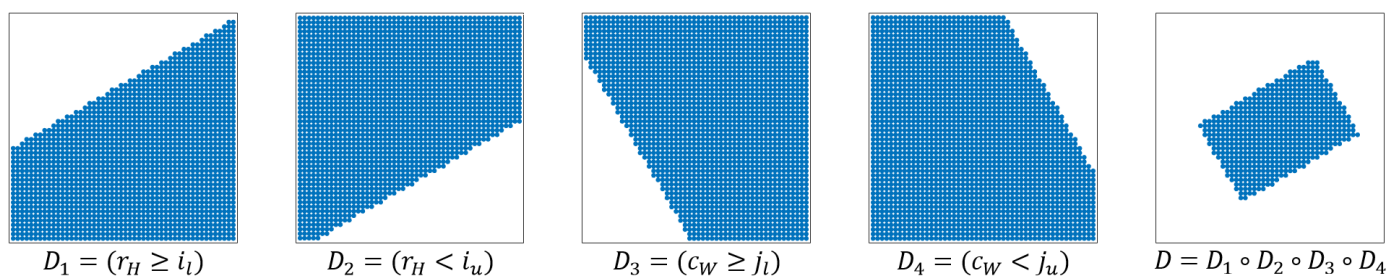


Figure 3. The constituent matrices (D_1 , D_2 , D_3 , D_4) and resulting representation (D) for an exemplary rectangle. The final object representation is attained by performing elementwise multiplication of four constituent matrices obtained by using simple vector inequalities.

For circles, the raster approximations can be obtained directly without the need to calculate multiple constituent matrices. Let r_1 be a row vector of ones with a length of H , c_1 be a column vector of ones with a length of W , and I_1 be an H by W matrix of ones. Then,

the distance matrices along rows (s_i) and columns (s_j), and resultant distance matrix (s) can be calculated as follows:

$$s_i = i_c I_1 - r_H C_1, \quad s_j = j_c I_1 - r_1 C_W, \quad s = \sqrt{s_i \circ s_i + s_j \circ s_j} \quad (5)$$

Finally, the matrix entities lying inside the circle can be filled with ones using the relation:

$$D(s \leq d/2) = 1 \quad (6)$$

3. Optimization Problem

3.1. Objective Functions

3.1.1. Total Ones (T1)

By denoting the number of matrix entities having the value k with N_k , the first objective function is simply defined as the total number of ones in the domain (N_1):

$$f_{T1} = N_1 \quad (7)$$

The maximization of this metric is equivalent to maximally occupying the design space without causing overlaps. However, this technique cannot be used for sequential placement as it would rank all feasible locations identically, thus creating an ill-posed problem.

3.1.2. Distance from the Center (DC)

As the second objective, the rectilinear (Manhattan) distance between the centers of the objects and domain is maximized. This approach favors filling the areas next to the boundaries starting from the corners. Let the row and column indices of the domain center be i_D^o and j_D^o , respectively. For a rectangle, the central row (i_r^o) and column (j_r^o) indices can be calculated as

$$i_r^o = i_r + h/2 \cos(\theta) - w/2 \sin(\theta), \quad j_r^o = j_r + h/2 \sin(\theta) + w/2 \cos(\theta) \quad (8)$$

Using the central indices, the distance function to be maximized is defined as

$$f_{DC}^r = |i_r^o - i_D^o| + |j_r^o - j_D^o| \quad (9)$$

For a circular object, the distance function can be written as follows:

$$f_{DC}^c = |i_c - i_D^o| + |j_c - j_D^o| \quad (10)$$

3.1.3. Distance to the Bottom-Left (DBL)

The third metric is defined as the rectilinear distance between the objects and the bottom-left corner of the domain. Minimization of this distance has been prevalently used in the previous packing studies [17,18]. The negative of the distance is chosen to be consistent with the other utilized metrics, which are all maximized in the optimization. Therefore, the objective functions for the rectangles (f_{DBL}^r) and circles (f_{DBL}^c) can be expressed as follows:

$$f_{DBL}^r = i_r - j_r, \quad f_{DBL}^c = i_c - j_c \quad (11)$$

This objective is only used with sequential packing.

3.1.4. Adjacent Ones (A1)

The number of adjacent ones is introduced as the last objective function, which correlates with the overall contact within the domain. This metric promotes efficient utilization of the available space and it is formulated by taking advantage of the raster representation. The objective function to be maximized (f_{A1}) is simply defined as the sum

of the entities of matrix F , which is the elementwise multiplication of four submatrices of the current domain matrix D :

$$F = D(1 : H - 1, 1 : W - 1) \circ D(2 : H, 1 : W - 1) \\ \circ D(1 : H - 1, 2 : W) \circ D(2 : H, 2 : W) \\ f_{A1} = \sum_{i=1}^{H-1} \sum_{j=1}^{W-1} F_{ij} \quad (12)$$

In Figure 4, the principle behind the developed contact maximization technique demonstrated for two example domains containing rectangles (a) in contact and (b) separated by zeroes.

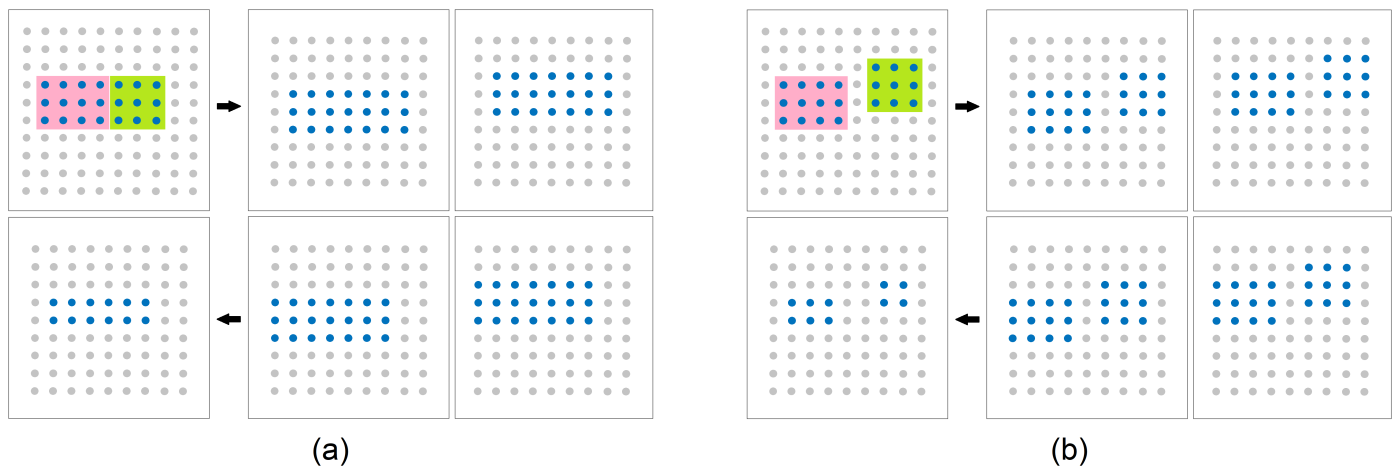


Figure 4. Demonstration of the principle behind maximizing adjacent ones using two sample domains that contain rectangles (a) in contact (b) separated by zeroes. The sparsity patterns for domain matrices, four submatrices, and elementwise multiplications of the submatrices are shown at the top-left corners, right sides, and bottom-left corners of the subfigures, respectively. The contacting rectangles produce 12 nonzero entities while the separated ones yield 10 nonzero entities.

The proposed adjacent ones metric provides several advantages:

- It outperforms other metrics in solving certain problems;
- It is versatile as it delivers satisfactory results for several different cases;
- It is applicable for both concurrent and ordered optimization strategies;
- It can be used with or without explicit constraints while performing well in many unconstrained cases.

3.2. Design Variables

The design variables considered within the optimization are the row and column indices for the upper-left corners of the rectangles (i_r 's and j_r 's) and centers of the circles (i_c 's and j_c 's). When the rotation is allowed for the rectangles, the angles of rotation (θ_r 's) are considered as additional design variables. For R rectangular and C circular objects, all the design variables can be assembled into a single vector x as follows:

$$x = [i_r^1 \quad j_r^1 \quad \theta_r^1 \quad \dots \quad i_r^R \quad j_r^R \quad \theta_r^R \quad i_c^1 \quad j_c^1 \quad \dots \quad i_c^C \quad j_c^C]^T \quad (13)$$

The domain of design variables is defined by the lower and upper bounds. For rectangular objects, the values of the upper left corner coordinates and the rotation angles are bounded as follows:

$$1 \leq i_r \leq H, \quad 1 \leq j_r \leq W, \quad -\pi/2 \leq \theta \leq \pi/2 \quad (14)$$

For circular objects, the bounds on the central indices are defined as

$$\begin{aligned}\lfloor d/2 \rfloor &\leq i_c \leq H - \lfloor d/2 \rfloor \\ \lfloor d/2 \rfloor &\leq j_c \leq W - \lfloor d/2 \rfloor\end{aligned}\quad (15)$$

3.3. Nonlinear Constraints

Two sets of nonlinear constraints are considered while solving the problem. The first set (g_1) ensures that the objects are contained within the design space, and the second set (g_2) enforces avoiding overlaps. These two sets are stacked together within a vector g that should be smaller than zero:

$$g(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \end{bmatrix} \leq 0 \quad (16)$$

3.3.1. Containment Assurance

When the rotational motion is permitted for the rectangular objects, the lower and upper bounds of the position variables do not guarantee the prevention of overflowing. Therefore, a nonlinear constraint set is defined as

$$g_1 = \begin{bmatrix} -i + w \sin(\theta) + 1 \\ i + h \cos(\theta) - H \\ -j + 1 \\ j + w / \cos(\theta) + (h - w \tan(\theta)) \sin(\theta) - W \end{bmatrix} \quad (17)$$

For circular objects, the containment is directly ensured through the limits imposed on the position variables.

3.3.2. Overlap Avoidance

When an object is placed into the domain, the matrices representing the new object and the domain with previous objects are simply summed together. An overlap is detected for the matrix elements with values higher than 1. Therefore, for O total objects, the constraint for the overlap avoidance can be written as

$$g_2 = \sum_{k=2}^O N_k \quad (18)$$

While maximizing total ones or adjacent ones as the performance metrics, explicit collision constraints may not be necessary since the improvement in the objective function and reduction in the overlaps are correlated in certain problems.

3.4. Optimizer

The described optimization problem is solved using a genetic algorithm capable of handling non-convex objectives. Specifically, commercial software MATLAB's 'ga' function is used with default options. The genetic algorithms have previously been proven effective for escaping locally optimal solutions arising in packing problems [18,19]. The design variables are constrained to be integers since the optimal positions should correspond to the matrix indices. In cases involving rotations, the orientation is controlled with the integer angle variable multiplied by the specified angular increment.

Besides their potency in circumventing local optimality, the genetic algorithms also avoid sensitivity-related problems of gradient-based methods. The sensitivities can be unstable in concurrent packing due to the repeated interference between multiple objects. In addition, the proposed f_{A1} metric is not compatible with the gradient-based optimizers as it often exhibits zero sensitivity to the perturbation of position variables.

4. Case Studies

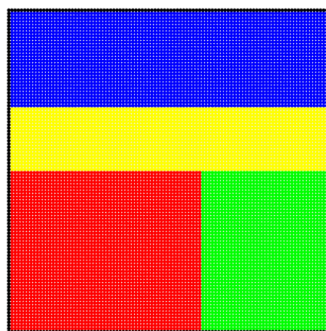
This section covers the results from different case studies that involved the packing of rectangular and circular objects into 100×100 square bins. Both concurrent and sequential optimization strategies were explored. In the sequential placement, the objects were ordered with the decreasing diagonal length or area. The total number of ones (f_{T1}), objects' rectilinear distances from the center (f_{DC}), objects' distance from the bottom-left corner (f_{DBL}), and the number of adjacent ones (f_{A1}) were used as the objective functions. The combined influence of the optimization settings on COMPACT's performance in solving different problems was investigated.

4.1. Non-Rotatable Objects

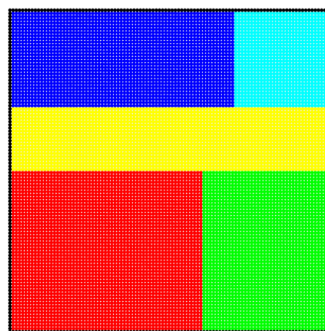
The developed methodology was initially utilized to obtain packing configurations for non-rotatable objects. The first study involved sets of rectangular objects whose heights and widths are given in Table 1. These sets were obtained by splitting one rectangle into two new ones at each step as demonstrated in Figure 5. Hence, the number of rectangles was regularly increased from 4 to 9 while going from the first case (a) to the last one (f).

Table 1. Heights and widths of the rectangles used in the sets used in the first study that involves no rotations.

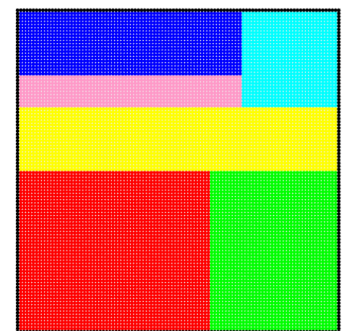
| Number of Objects | (Height, Width) | | | | | | | | |
|-------------------|-----------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| 4 | (50, 60) | (50, 40) | (30, 100) | (20, 100) | | | | | |
| 5 | (50, 60) | (50, 40) | (30, 70) | (30, 30) | (20, 100) | | | | |
| 6 | (50, 60) | (50, 40) | (20, 70) | (10, 70) | (30, 30) | (20, 100) | | | |
| 7 | (50, 60) | (50, 40) | (20, 70) | (10, 70) | (30, 20) | (30, 10) | (20, 100) | | |
| 8 | (50, 60) | (50, 40) | (20, 70) | (10, 70) | (30, 20) | (30, 10) | (15, 100) | (5, 100) | |
| 9 | (50, 60) | (50, 40) | (15, 70) | (5, 70) | (10, 70) | (30, 20) | (30, 10) | (15, 100) | (5, 100) |



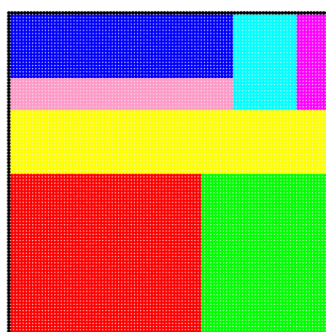
(a)



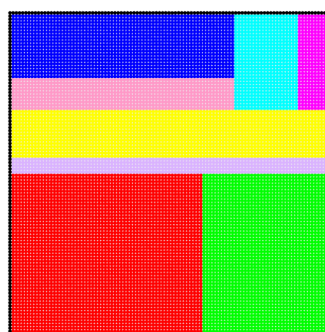
(b)



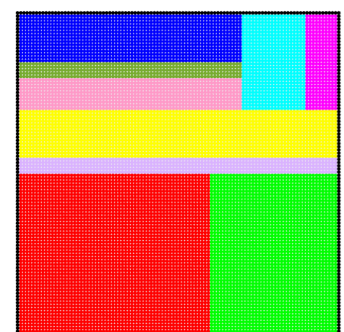
(c)



(d)



(e)



(f)

Figure 5. Sample optimal solutions for different sets of rectangles used in the first study with restrained rotations. When packed optimally the objects perfectly fill the domain and their number in the sets increases from (a–f) as one rectangle is split into two new ones at each step.

Table 2 shows the success rates in packing all objects of each set for different optimization strategies and objective functions. For each case, the average results for 100 computations are presented. In the concurrent optimization, the effectiveness of maximizing total ones decreased with the increasing number of objects. Maximizing adjacent ones showed a similar trend but gave slightly better results for the higher number of objects. No collision avoidance constraints were considered in these studies since the higher values of the objectives also favor reducing the overlaps.

For the sequential optimization, the optimal solutions were found for all trials for the sets with four and five objects. For the analyzed sets, sorting by diagonal length generally performed better compared to area sorting, which was more slightly more effective for only intermediate numbers (6, 7) of objects. For the length sorting, using the distance to the bottom-left corner as the objective function provided the best results. However, maximizing adjacent ones also worked fairly well providing a 75% success rate for 9 objects.

Table 2. Success percentages in packing all objects of each set given in Table 1 for different optimization strategies and objective functions.

| Optimization Strategy | | Objective Function | Number of Objects | | | | | |
|-----------------------|-------------------------------|--------------------|-------------------|------|------|------|------|-----|
| | | | 4 | 5 | 6 | 7 | 8 | 9 |
| Concurrent | | f_{T1} | 100% | 84% | 51% | 42% | 23% | 5% |
| | | f_{A1} | 100% | 78% | 42% | 46% | 27% | 9% |
| Ordered | Diagonal length-wise ordering | f_{DC} | 100% | 100% | 41% | 49% | 45% | 20% |
| | | f_{DBL} | 100% | 100% | 99% | 99% | 100% | 94% |
| | | f_{A1} | 100% | 100% | 81% | 81% | 73% | 75% |
| | Area-wise ordering | f_{DC} | 100% | 100% | 100% | 51% | 6% | 7% |
| | | f_{DBL} | 100% | 100% | 100% | 100% | 13% | 24% |
| | | f_{A1} | 100% | 100% | 98% | 100% | 9% | 6% |

For the second study, another set of objects with the quantities and dimensions specified in Table 3 was used. This set perfectly occupies the domain for optimal packing, which requires placement of the objects according to a regular pattern. For concurrent optimization, the optimal packing could not be found using f_{T1} as the performance metric, where f_{A1} still succeeded 6 times out of 100 trials. For sequential placement, f_{A1} delivered the best results by far with its 40% success rate, where f_{DC} and f_{DBL} succeeded in 7% and none of the trials, respectively. The effectiveness of f_{A1} originates from its inclination towards maximizing the line of contact, which in turn aids the formation of the required pattern. In Figure 6, (a) one optimal solution for the second study and (b) positioning preferences of f_{DC} and f_{A1} during a sample run are demonstrated.

Table 3. Quantities and dimensions of the rectangles used in the second study that involves no rotations.

| Quantity | Height | Width |
|----------|--------|-------|
| 2 | 80 | 20 |
| 2 | 20 | 80 |
| 2 | 40 | 20 |
| 2 | 20 | 40 |
| 1 | 20 | 20 |

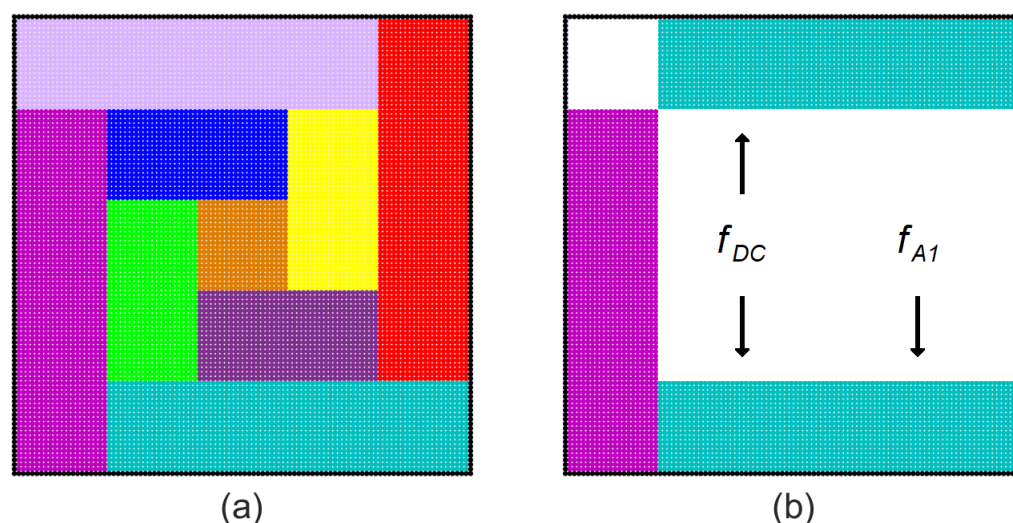


Figure 6. (a) One optimal solution for the objects used in the second study with restrained rotations. The objects perfectly fill the domain for optimal packing. (b) Demonstration of two possible positions for the placement of the second object during a sample run. For f_{DC} metric both positions are equally preferable whereas the bottom location is more preferable for f_{A1} .

4.2. Rotatable Objects

Objects with large aspect ratios may require alteration of their orientation to be fit into the available space. New sets of objects were generated to test the developed methodologies on the packing problems with rotational degrees of freedom. The orientation angle was restricted to be a multiple of 45° to cover diagonal placement possibilities while keeping the problem size minimal.

In the first investigation, a (45×45) square and four smaller circles of 39 units diameter were packed into the square domain. Figure 7 shows the optimal packing configuration for these objects. Despite the relatively simple characteristics of the problem, the optimal solution cannot be attained with the ordered placement based on decreasing size.

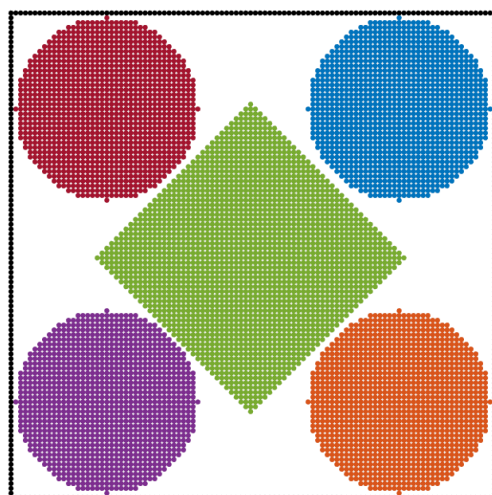


Figure 7. The optimal solution for the objects used in the first study involving rotational degrees of freedom.

The reason is that all three performance metrics (f_{DC} , f_{DBL} , and f_{A1}) tend to position the large square on a corner leaving room for only three of the circles. Only using f_{DC} with the small-to-big ordering leads to the optimal solution indicating that packing smaller objects first can be advantageous in certain problems. While the sequential placement performed ineffectively for this problem, concurrent placement with active constraints

provided success rates of 5% and 7% for f_{T1} and f_{A1} metrics, respectively. Therefore, if the problem size permits its use, the concurrent optimization strategy can offer versatility in tackling different problems. As another investigation, the concurrent optimization cases were repeated by removing collision and containment constraints. In the absence of constraints, f_{T1} and f_{A1} still yielded 5% and 4% success rates, respectively. These results demonstrate that these metrics intrinsically facilitate the containment of rotatable objects in addition to collision avoidance.

The second case study concerned packing three rectangles and six circles into a square domain. The quantities and dimensions of the objects used in this study are given in Table 4. With the optimal packing, all objects in the set can be placed into the domain providing a filling ratio of 74.4% as shown in Figure 8.

The concurrent optimization strategy was not effective for packing this set due to the increased number of objects and extra rotational degrees of freedom. For this problem, the filling ratios were compared for f_{DC} , f_{DBL} , and f_{A1} metrics while collision and containment constraints were held active. When the objects were ordered in decreasing length (diagonal length for the rectangles and diameter for the circles) the average filling ratios over 100 trials were found as 56.7% and 67.7% using f_{DC} and f_{DBL} , respectively. For this ordering, the best results were achieved with f_{A1} , which yielded an average filling ratio of 70.8%. In this case, f_{A1} outperformed the other metrics since it prefers positioning the smaller rectangles right next to the large rectangle, which is diagonally packed in the first step. Such placement leaves sufficient room at the corners where the large circles can be packed. When area-wise sorting was used, f_{DC} , f_{DBL} , and f_{A1} provided 66.3%, 60.4%, and 56.5% mean filling ratios, respectively. Hence, the relative effectiveness of the utilized metrics can be significantly influenced by the ordering strategy.

Table 4. Quantities and dimensions of the rectangular and circular objects used in the second study involving rotational degrees of freedom.

| Rectangle | | | Circle | |
|-----------|--------|-------|----------|----------|
| Quantity | Height | Width | Quantity | Diameter |
| 1 | 10 | 120 | 2 | 39 |
| 2 | 10 | 90 | 4 | 19 |

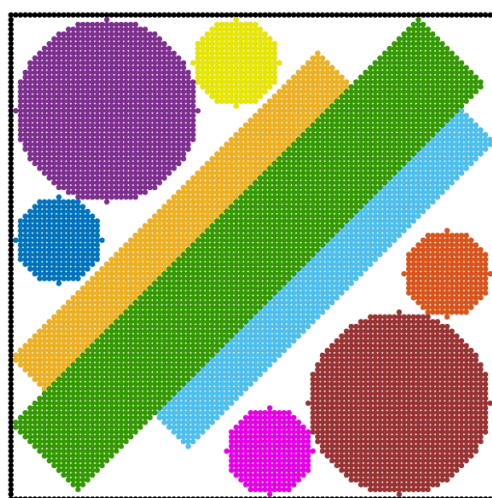


Figure 8. A sample optimal solution for the objects used in the second study involving rotational degrees of freedom. The filling ratio for this packing configuration is 74.4%.

In the last study, the number of objects was increased to create a more difficult problem, where the quantities and dimensions of the packed objects are specified in Table 5. One possible optimal packing configuration for these objects is depicted in Figure 9, where the

filling ratio is 92.3%. Note that, length-wise and area-wise ordering strategies are identical for this set.

For this problem, f_{DC} , f_{DBL} , and f_{A1} provided average filling ratios of 87.2%, 84.3%, and 82.2%, respectively. In this case, slight superiority of f_{DC} is due to its preference of occupying the corners with the largest objects, which allows the intermediate rectangle to be diagonally placed into space in the middle. The configuration with the highest filling ratio (89.8%) was also achieved by f_{DC} . The f_{DBL} metric provided the second-highest mean ratio and showed almost no variability over the runs. Although f_{A1} gave the lowest mean filling ratio, it showed greater variability compared to f_{DBL} and provided a maximum value of 88.2%, which is higher than the maximum ratio obtained by f_{DBL} .

Table 5. Quantities and dimensions of the rectangular and circular objects used in the third study involving rotational degrees of freedom.

| Rectangle | | | Circle | |
|-----------|--------|-------|----------|----------|
| Quantity | Height | Width | Quantity | Diameter |
| 4 | 40 | 40 | 2 | 19 |
| 1 | 28 | 28 | 4 | 11 |
| 4 | 20 | 15 | | |

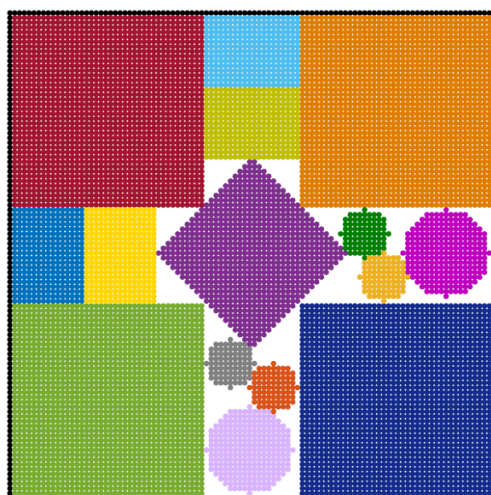


Figure 9. A sample optimal solution for the objects used in the third study involving rotational degrees of freedom. The filling ratio for this packing configuration is 92.3%.

5. Conclusions

This study presents the new Concurrent or Ordered Matrix-based Packing Arrangement Computation Technique (COMPACT). The proposed method relies on approximating the object shapes by matrix-based (raster) representations. Such an approach provides versatility for packing various shapes and allows the use of new performance metrics.

In the proposed framework, the raster representations are obtained through loop-free operations that improve efficiency. The objects can be rotated by arbitrary angles, different from the right-angled rotation restrictions imposed in many existing packing optimization studies based on raster methods. Besides, a novel performance metric is introduced. This metric facilitates efficient filling of the available space by maximizing the overall contact within the domain. Moreover, exploitation of the objective functions for discarding overlap and overflow constraints is explored to enable the use of unconstrained optimization methods.

As case studies, the developed technique was utilized to pack multiple rectangular and circular objects in square bins. The suitability of concurrent and ordered placement strategies for solving different problems was investigated. In addition, the effectiveness of

the utilized performance metrics was analyzed for various scenarios. The results showed that the proposed technique performs effectively in obtaining plausible packing configurations. The case studies also provide guidelines for the selection of suitable packing strategies, performance metrics, and constraints according to the problem type.

The developed techniques can certainly be extended for three-dimensional objects in future studies. They can also be used for packing problems involving various other objects and domain geometries. In particular, the new performance metric relying on contact maximization offers great potential for packing non-convex polygons. Although the genetic algorithms perform well with the described formulations, the implementation of other nonlinear optimization methods into the developed framework is also possible.

Funding: This research received no external funding. The article processing charge was covered by the Max Planck Digital Library.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the author.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|-----|-----------------------------|
| A1 | Adjacent ones |
| DBL | Distance to the bottom-left |
| DC | Distance from the center |
| T1 | Total ones |

References

- Chen, M.; Huang, W. A two-level search algorithm for 2D rectangular packing problem. *Comput. Ind. Eng.* **2007**, *53*, 123–136. [\[CrossRef\]](#)
- Leung, S.C.H.; Zhang, D.; Zhou, C.; Wu, T. A hybrid simulated annealing metaheuristic algorithm for the two-dimensional knapsack packing problem. *Comput. Oper. Res.* **2012**, *39*, 64–73. [\[CrossRef\]](#)
- Shiangjen, K.; Chaijaruwanich, J.; Srisujalertwaja, W.; Unachak, P.; Somhom, S. An iterative bidirectional heuristic placement algorithm for solving the two-dimensional knapsack packing problem. *Eng. Optim.* **2018**, *50*, 347–365. [\[CrossRef\]](#)
- Chen, M.; Wu, C.; Tang, X.; Peng, X.; Zeng, Z.; Liu, S. An efficient deterministic heuristic algorithm for the rectangular packing problem. *Comput. Ind. Eng.* **2019**, *137*, 106097. [\[CrossRef\]](#)
- Li, Y.; Sang, H.; Xiong, X.; Li, Y. An Improved Adaptive Genetic Algorithm for Two-Dimensional Rectangular Packing Problem. *Appl. Sci.* **2021**, *11*, 413. [\[CrossRef\]](#)
- Del Valle, A.M.; de Queiroz, T.A.; Miyazawa, F.K.; Xavier, E.C. Heuristics for two-dimensional knapsack and cutting stock problems with items of irregular shape. *Expert Syst. Appl.* **2012**, *39*, 12589–12598. [\[CrossRef\]](#)
- Chen, M.; Tang, X.; Song, T.; Zeng, Z.; Peng, X.; Liu, S. Greedy heuristic algorithm for packing equal circles into a circular container. *Comput. Ind. Eng.* **2018**, *119*, 114–120. [\[CrossRef\]](#)
- Bouazid, M.C.; Salhi, S. Packing Rectangles into a Fixed Size Circular Container: Constructive and Metaheuristic Search Approaches. *Eur. J. Oper. Res.* **2020**, *285*, 865–883. [\[CrossRef\]](#)
- Cherri, L.H.; Mundim, L.R.; Andretta, M.; Toledo, F.M.B.; Oliveira, J.F.; Carravilla, M.A. Robust mixed-integer linear programming models for the irregular strip packing problem. *Eur. J. Oper. Res.* **2016**, *253*, 570–583. [\[CrossRef\]](#)
- Zhang, W.; Zhang, Q. Finite-circle method for component approximation and packing design optimization. *Eng. Optim.* **2009**, *41*, 971–987. [\[CrossRef\]](#)
- Martins, T.C.; Tsuzuki, M.S.G. Simulated annealing applied to the irregular rotational placement of shapes over containers with fixed dimensions. *Expert Syst. Appl.* **2010**, *37*, 1955–1972. [\[CrossRef\]](#)
- Martinez-Sykora, A.; Alvarez-Valdes, R.; Bennell, J.A.; Ruiz, R.; Tamarit, J.M. Matheuristics for the irregular bin packing problem with free rotations. *Eur. J. Oper. Res.* **2017**, *258*, 440–455. [\[CrossRef\]](#)
- Leao, A.A.S.; Toledo, F.M.B.; Oliveira, J.F.; Carravilla, M.A.; Alvarez-Valdés, R. Irregular packing problems: A review of mathematical models. *Eur. J. Oper. Res.* **2020**, *282*, 803–822. [\[CrossRef\]](#)
- 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.* **2013**, *145*, 478–487. [\[CrossRef\]](#)

15. Mundim, L.R.; Andretta, M.; de Queiroz, T.A. A biased random key genetic algorithm for open dimension nesting problems using no-fit raster. *Expert Syst. Appl.* **2017**, *15*, 358–371. [[CrossRef](#)]
16. Sato, A.K.; Martins, T.C.; Gomes, A.M.; Tsuzuki, M.S.G. Raster penetration map applied to the irregular packing problem. *Eur. J. Oper. Res.* **2019**, *279*, 657–671. [[CrossRef](#)]
17. Hopper, E.; Turton, B. A Genetic Algorithm for a 2D Industrial Packing Problem. *Comput. Ind. Eng.* **1999**, *37*, 375–378. [[CrossRef](#)]
18. Jakobs, S. Theory and Methodology On genetic algorithms for the packing of polygons. *Eur. J. Oper. Res.* **1996**, *88*, 165–181. [[CrossRef](#)]
19. Hopper, E.; Turton, B.C.H. An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *Eur. J. Oper. Res.* **2001**, *128*, 34–57. [[CrossRef](#)]