New results for several rectangle packing and covering problems

Lower bounds and different types of algorithms

Tomas Attila Olaj

Short version of the dissertation submitted to the Doctoral School of Information Science and Technology for the degree of Doctor of Philosophy (PhD)

PhD Thesis in Computer Science, with a focus on Combinatorial Optimization

Supervisor: Dr. György Dósa

University of Pannonia
Faculty of Information Technology
Department of mathematics
Doctoral School of Information Science and Technology

2025, Hungary

1 Introduction

This thesis is divided into two main parts. The first introduces the Board Packing Problem (BoPP), a combined optimization problem where rectangles are allocated on a given board subject to specific rules. The second part addresses square packing and square covering problems for a special, yet interesting, case where the instance of squares consists of one square of size 1, one of size 2, and so on, up to one square of size n, where n is a fixed integer. Before presenting these contributions, a brief overview of packing problems is provided.

Packing problems appear in many areas of science and industry, including manufacturing, storage and warehousing, transportation, finance, and construction. When solved optimally, they can lead to cost savings and improved use of resources. A packing problem consists of fitting objects of various sizes and/or shapes into given spaces while minimizing waste or maximizing profit or efficiency. For an extensive introduction, see [22].

The simplest variation is the one-dimensional bin packing problem. It consists of assigning n items with different weights to identical bins of fixed capacity, such that the number of bins used is minimized. This is a classical optimization problem and is $\mathcal{NP}$-hard [22], which implies that no algorithm is known to find optimal solutions in polynomial time. Since the one-dimensional case is $\mathcal{NP}$-hard, all of its generalizations are also $\mathcal{NP}$-hard.

Because of this complexity, finding an exact solution in reasonable time is generally infeasible for large instances. Instead, heuristic algorithms and approximation techniques are commonly applied to compute good, though not necessarily optimal, solutions efficiently.

Lodi et al. [19] provide a detailed survey on two-dimensional bin packing, reviewing exact, heuristic, and approximation algorithms for both rotational and orthogonal variants. Caprara et al. [9] focus on the orthogonal two-dimensional case.

2 The Board Packing Problem (BoPP)

Consider a rectangular board with m rows and n columns. Let $M = \{1, \ldots, m\}$ denote the set of rows and $N = \{1, \ldots, n\}$ the set of columns. Each position $(i, j) \in M \times N$ of the board is associated with an integer value $g_{i,j}$ representing the revenue obtained if the position is covered.

A set R of rectangles is given, where each rectangle $r \in R$ has a specified height h_r, width w_r, and cost c_r. The objective is to select and place rectangles on the board to maximize the profit, defined as the sum of the revenues of the covered positions minus the costs of the purchased rectangles. Rectangles must be placed with sides parallel to the sides of the board. Overlaps are permitted, but the revenue from a given board position can be collected at most once. This formulation defines the *Board Packing Problem* (BoPP).

The BoPP was first introduced by Dosa et al. [12], where the special case without overlaps was investigated. That initial publication presented a mathematical model and applied a commercial mixed-integer programming solver (CPLEX) to obtain optimal solutions for a small set of test instances. The problem was subsequently studied in greater detail in [1], and further investigations are reported in [2].

2.1 Application, and related problems

An application of the BoPP arises when the board represents a geographical area, such as a city or a country. The rectangles represent facilities that can be constructed at certain positions, and the covered g values represent the profit obtained from the operation of these facilities.

- If $g_{i,j}$ is negative, it is preferable not to cover the point, for example because the location is already inhabited or owned by a person or company. If it is nevertheless covered, a penalty or expropriation cost must be paid.
- The purchasing cost $c_r \geq 0$ of rectangle r represents the installation cost of a new facility.

The BoPP is closely related to other problems such as the rectangle blanket problem [10], which has applications in computer vision, including template matching and people tracking.

Allowing rectangles to overlap in the BoPP is motivated by applications where rectangular resources can be purchased and applied on a grid to obtain benefits. For instance, the rectangles may represent satellite images [17] that can be acquired to extract useful or profitable information about geographical regions, with applications ranging from post-disaster damage assessment [14], highway road maintenance [15], to the detection of archeological features [21]. Demiröz [10] also discussed problem variants within covering and partitioning, as well as cutting and packing, to highlight their connections to the rectangle blanket problem.

Several studies consider cases where partial covering of customer demands is possible (see, e.g., Berman and Kras [7] or Blanquero et al. [8]). The focus here is restricted to models in which a demand point is either completely served or not served. Related problems have also been studied extensively in the literature (see, e.g., [18]).

2.2 $\mathcal{NP}$-hardness of Board Packing

The $\mathcal{NP}$-hardness of the BoPP can be established in several ways. A standard approach is to demonstrate a reduction from other $\mathcal{NP}$-hard problems. The following shows how the hardness of the BoPP can be proven through such reductions.

2.2.1 Reduction from other $\mathcal{NP}$-hard packing models

The $\mathcal{NP}$-hardness of the BoPP can be shown for the following special case.

2.2.1.1 Reduction from the model of Masek [20] Consider the case where the g values may be negative, with very large absolute values. In the construction, g is restricted to take only two values: a fixed small positive value (here, 2) and a negative value with large absolute magnitude.

Suppose that K (the set of rectangles) is part of the input, and that overlap is allowed. Given an instance of the rectilinear covering problem of Masek (where a board is defined by entries of 1 and 0, all 1's must be covered, and no 0 may be covered), a corresponding BoPP instance is created.

In this construction, a cell is called a *good point* if its value in the Masek model is 1, and a *bad point* if its value is 0. The Masek board is embedded into a BoPP board of size $M \times N$, with $p = \max(M, N)$. Set $g(i, j) = 2$ for good points (i, j) and $g(i, j) = -2p^2$ for bad points and for points outside the board.

The set of rectangles is then defined as follows: for every $1 \leq i \leq M$ and $1 \leq j \leq N$, include an $i \times j$ rectangle, and create $M \cdot N$ copies of each. The cost of each rectangle is set to 1.

It follows that any optimal cover in the Masek model corresponds to an optimal cover in the BoPP. In the Masek model, all 1's are covered, no 0's are covered, and the minimum number of rectangles is used.

In the BoPP model, no rectangle extends outside the board in an optimal solution, as this would be unprofitable. Similarly, no bad point is covered. Thus, any optimal solution in the BoPP covers only good points. If any good point were left uncovered, it would always be profitable to purchase a 1×1 rectangle (cost 1, revenue 2) to cover it. Therefore, all good points are covered.

Since each rectangle incurs a positive cost, an optimal BoPP solution will use a minimum number of rectangles to cover all good points, matching the solution of the Masek model.

This shows that for any instance of Masek's problem, a corresponding BoPP instance can be constructed where the optimal solution covers the same points with the same number of rectangles. Hence, the BoPP is at least as hard as the rectilinear covering problem of Masek, proving $\mathcal{NP}$-hardness by reduction.

2.2.1.2 Reduction from the red-blue points model of Bereg et al. [6] Another reduction can be obtained from the red-blue points model of Bereg et al. [6]. The reduction follows a similar approach as in the case of Masek. In this model, there are *good* points and *bad* points, and the objective is to cover all good points while avoiding the coverage of any bad point, using the minimum number of rectangles. This problem is $\mathcal{NP}$-hard, even in the special case where the rectangles are restricted to be squares.

To construct the BoPP instance, assign a gain value of $-2p^2$ to each bad point, a gain value of 2 to each good point, and a gain value of 0 to any point that is neither good nor bad in the Bereg model. With these assignments, the remainder of the reduction proceeds in the same way as in the reduction from Masek's model.

2.2.1.3 Reduction from $\mathcal{NP}$-hard SAT problems Another approach is to prove directly, without relying on other packing models, that the BoPP is $\mathcal{NP}$-hard, even in highly restricted cases. This can be shown through reductions from appropriate SAT problems.

Formally, consider the following problem, where h and w are positive integers and S is a given set of integers. The set S defines the possible gain values $g_{i,j}$, while the purchasable rectangles $r \in R$ are restricted to a single size $h \times w$.

BoardPacking$(h, w; S)$ (BP$(h, w; S)$)
Input: An $M \times N$ rectangular board (matrix) with all entries from S, and an integer g.
Question: Does there exist a packing with at most $M \cdot N$ rectangles of size $h \times w$, each having a cost of 1, with total profit at least g?

Two restricted versions of this problem are of particular interest:
BinaryBoardPacking(h, w) (BBP(h, w)): BP$(h, w; S)$ with $S = \{0, 1\}$,
AlmostBinaryBoardPacking(h, w) (ABP(h, w)): BP$(h, w; S)$ with $S = \{-1, 0, 1\}$.

Theorem 1. *The following problems are $\mathcal{NP}$-complete, and their optimization versions are $\mathcal{NP}$-hard:*

(i) BBP$(3, 3)$,

(ii) ABP$(2, 3)$ and ABP$(3, 2)$,

(iii) BBP$(2, 2)$.

The proofs are based on reductions from variants of the Boolean Satisfiability problem. In general, given a satisfiability instance with p clauses over a set of q variables, the derived instance matrix (board) for BBP or ABP has size $O(p) \times O(q)$, and constructing such an instance requires $O(pq)$ time. Hence, the reduction is quadratic in both time and space.

2.3 Mathematical programming model

A binary integer programming model for the BoPP can be formulated as follows. Let A_r denote the set of feasible coordinates (i, j) for the top-left corner of rectangle $r \in R$. Define binary variable x_{ijr} to take the value 1 if and only if rectangle r is placed with its top-left corner at (i, j). Let B_{ijr} denote the set of positions (u, v) for the top-left corner of rectangle r that result in position (i, j) being covered by that rectangle.

To calculate the revenues from covered board positions, an additional binary variable y_{ij} is introduced, which takes the value 1 if and only if position (i, j) of the board is covered. The model can then be written as:

$$\max \sum_{i \in M} \sum_{j \in N} g_{ij} y_{ij} - \sum_{r \in R} \sum_{(i,j) \in A_r} c_r x_{ijr} \tag{1}$$

$$\sum_{(i,j) \in A_r} x_{ijr} \leq 1, \qquad r \in R \tag{2}$$

$$y_{ij} \leq \sum_{r \in R} \sum_{(u,v) \in B_{ijr}} x_{uvr}, \qquad i \in M, j \in N : g_{ij} > 0 \tag{3}$$

$$|R| y_{ij} \geq \sum_{r \in R} \sum_{(u,v) \in B_{ijr}} x_{uvr}, \qquad i \in M, j \in N : g_{ij} < 0 \tag{4}$$

$$x_{ijr} \in \{0, 1\}, \qquad r \in R, (i, j) \in A_r \tag{5}$$

$$y_{ij} \in \{0, 1\}, \qquad i \in M, j \in N \tag{6}$$

The objective function (1) computes the total gain by summing the values of all covered positions, allowing for overlap, while subtracting the total cost incurred by the rectangles used to achieve this coverage. Constraints (2) and (5) ensure that each rectangle is used at most once. Constraints (3) and (6) guarantee that y_{ij} can take the value 1 only if position (i, j) is covered by at least one rectangle when $g_{ij} > 0$. Constraints (4) and (6) ensure that y_{ij} must take the value 1 if position (i, j) is covered by any rectangle and $g_{ij} < 0$.

Constraints (3)–(4) together introduce a total of $|M||N|$ restrictions. Although both sets of constraints are defined for all board positions (i, j), the maximization of the objective function implies that half of them are automatically satisfied in any optimal solution of the model.

2.4 Evolutionary algorithm

An evolutionary algorithm was implemented to provide heuristic solutions, capable of handling instances that are too large to be solved to optimality using exact methods. The heuristic maintains a set of solutions in memory, referred to as a population. New solutions are generated by combining pairs of existing solutions. Local improvements are performed on the new solutions before they are added to the population. When the population reaches a certain size, it is trimmed down to a target size by selecting the best solutions from the full population. If the search has trimmed down the population a certain number of times without finding a new best solution in between, the search is restarted from scratch. Figure 1 illustrates the components of the search.

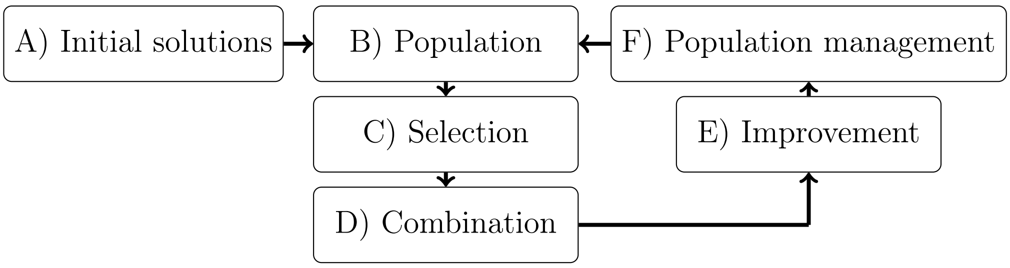

Figure 1: An overview of the evolutionary algorithm developed for the BoPP.

Several refinements were incorporated into the evolutionary algorithm to improve its efficiency.

2.5 Computational experiments

Although the problem under study is $\mathcal{NP}$-hard, the practical difficulty of finding optimal solutions is not fully understood. The computational experiments therefore aim to evaluate the ability of a commercial mixed-integer programming solver to identify optimal solutions and provide proofs of optimality. In addition, the performance of the proposed evolutionary algorithm is studied, both in terms of its ability to find near-optimal heuristic solutions for instances with known optimal solutions, and its ability to obtain good feasible solutions in cases where the commercial solver cannot be applied successfully. The objective is not to demonstrate that one solution method is superior to the other, but rather to investigate the instance characteristics that allow each method to exhibit superior performance and to gain insights into how these characteristics affect overall performance.

In the experiments, CPLEX (version 20.1) is used as the commercial solver with a time limit of 600 seconds. The evolutionary algorithm is executed with a time limit of 60 seconds. Both methods are run on a desktop computer with an Intel Core i3-4150 (Dual-Core) CPU at 3.50 GHz and 8.00 GB of RAM under Windows 10 Pro 21H1. However, to validate the observed performance ceiling, it was also performed tests on a high-end MacBook Pro with 32 GB of RAM and faster CPU architecture, using extended time limits and CPLEX parameter tuning.

The following investigations are conducted:

1. the effect of increasing the board size, while keeping other aspects of the problem fixed;
2. the effect of the number of rectangles;
3. the effect of the variety of different rectangles;
4. the effect of the board topography, i.e., how the profits of covering board positions are related to each other;
5. performance on artificial instances based on covering a given landmass with satellite images;
6. performance on sets of instances that are constructed to be very similar to each other.

A detailed evaluation of the results is provided in the thesis.

3 Square Packing and Covering

3.1 The problems under investigation

Consider the following setting: one square of size 1, one square of size 2, and so on, up to one square of size n, where n is a natural number. These squares are referred to as *items*. The problem is to determine the smallest square of size K into which all items fit, without rotation (the sides of the items must remain parallel to the sides of the accommodation square) and without overlap. This question corresponds to Problem A005842 in the Online Encyclopedia of Integer Sequences [24].

Gardner [13] presented the best possible (tight) results for this problem up to $n = 17$, based on earlier work. An upper bound (UB) is best possible if it coincides with a valid lower bound (LB).

A review of the current best upper bounds for the packing problem is given in [3], and the results are also available on the webpage [24]. There, the best known $UB(n)$ values are listed up to $n = 56$ (most of them tight), where $UB(n)$ denotes the upper bound for the packing problem with a given n. One of the strongest lower bounds is derived from the total area of the squares, rounded up by taking the square root, i.e. $LB_0 := \left\lceil \sqrt{\sum_{k=1}^{n} k^2} \right\rceil = \left\lceil \sqrt{\frac{n(n+1)(2n+1)}{6}} \right\rceil$. Tightness (i.e., when UB equals a particular LB) has not been proved for $n = 38, 40, 42, 48, 52, 53,$ and

55, while for all other values with $1 \le n \le 56$ the values of $UB(n)$ reported in [24] are proved to be tight.

The latest results were obtained by Hougardy [16], who showed that $UB(28) = 89$, $UB(32) = 108$, $UB(33) = 113$, $UB(34) = 118$, and $UB(47) = 190$, and proved that these upper bounds are best possible. In these cases, $LB_0 + 1$ serves as a matching lower bound.

Several lower bounds for the packing problem are presented in the literature, along with explicit packings. It has been established that for $1 \le n \le 24$ (except $n = 18$ and $n = 24$), the tightness of the best known packing can be validated by a simple lower bound. The case $n = 24$ was recently resolved in [23], where a complete combinatorial proof was provided. For $n = 18$, however, the problem remains open: computational search shows that the squares $[1, 18]$ cannot be packed into a square of size 46, but they can be packed into a square of size $K = 47$. Thus, $K = 47$ is known to be the optimal value, although the underlying reason is not yet understood.

In [3], the first asymptotic results for the packing problem are given. It is shown that there exists a constant $c < 1$ such that a square of size $N+cn$ admits a guillotine-type packing of the squares of sizes $[1, n]$, where $N := \sqrt{A_n} = \sqrt{\frac{n(n+1)(2n+1)}{6}}$.

3.2 Small and medium cases

Gardner [13] presented the best possible (tight) results up to $n = 17$, based on earlier works. The case $n = 2$ is trivial. For $n = 3$, the two largest squares must fit next to each other, so the accommodating square must have size at least $2 + 3$. This property holds for larger instances as well, which yields a valid lower bound for a packing.

Claim 1. *The following value is a lower bound for the size of the accommodation square:*

$$LB_1 = n + (n - 1).$$

The following lemma also holds:

Lemma 1. *For any set of five squares, at least three are collinear.*

Corollary 1. *The following value is a lower bound for the size of the accommodation square:*

$$LB_2 = (n - 2) + (n - 3) + (n - 4).$$

It can be observed that LB_2 is a tight lower bound for $n = 9, \ldots, 16$, but not for larger values.

3.2.1 All cases up to $n = 24$ and beyond

The results for $n = 2, \ldots, 24$ are summarized as follows. For $n = 2, \ldots, 8$, LB_1 is tight, and this lower bound is the only one that is tight, except that LB_0 is also tight for $n = 3$ and $n = 8$, and LB_2 is tight for $n = 8$.

For $n = 9, \ldots, 16$, LB_2 is tight, and this lower bound is the only one tight, except that LB_0 is also tight for $n = 15$ and $n = 16$. After $n = 16$, LB_2 is no longer tight.

For $n = 17, \dots, 24$, LB_1 is already far from the tight value. When stepping from n to $n+1$, LB_2 always increases by 3, while LB_0 first increases by 3, and then by 4 in this range of values. Thus, after $n = 16$, LB_2 cannot be competitive.

For the cases $n = 25, \dots, 56$, the evaluation is provided in the thesis. Here, LB_0 is sometimes tight and sometimes not, but the following observations can be made:

- If LB_0 is not tight, the difference from the tight value is only 1.
- Considering the range $2 \leq n \leq 56$, it can be observed that LB_0 ceases to be tight after a certain value of n.

3.3 An asymptotically optimal algorithm by Guillotine cutting

To the best of current knowledge, no asymptotic-type guarantees exist for the problem of packing consecutive squares into a square. This section provides the first results of this type, showing that a square of side length $N + cn$ admits a guillotine-type packing, where $c < 1$ is a constant and

$$N := \sqrt{\frac{n(n+1)(2n+1)}{6}}.$$

The details of the constructive phases are omitted here, as the main point is to establish the asymptotic bound. It is sufficient to note that the construction relies on recursive guillotine cuts that partition the square while maintaining feasibility. This yields the following result.

Theorem 2. *There exists a constant $c < 1$ such that the square of size $N + cn$ admits a guillotine-type packing of the squares of sizes $1, 2, \dots, n$.*

3.4 The covering problem

Now the covering problem is introduced by considering the following question:

Question: *Can a square of size $K \times K$ be completely covered by the consecutive squares of sizes $[1, n]$? In other words, what is the largest K for which the consecutive squares $[1, n]$ can fully cover a $K \times K$ square when allocated carefully?*

Overlaps among the small squares are permitted. To the best of current knowledge, this question has not previously been investigated. The main results for the covering problem can be summarized as follows:

- For small values of n, the covering problem is easy, and optimal solutions can be obtained. These can be determined either by hand or by a mathematical programming solver. Several formulations are possible, and two alternative models are discussed.
- For moderate values of n, such as $n = 23$, the problem becomes computationally hard for mathematical programming solvers. In these cases, a heuristic algorithm is applied that can find near-optimal solutions also for larger instances.

- An expansion algorithm is provided which, starting from a good cover of $K \times K$ for some n, can generate a cover for a larger square $K' > K$ using the small squares up to and including $n+1$.
- A simple covering policy is shown to yield an asymptotically optimal covering.

3.5 Small and medium cases for covering

Small cases up to $n = 7$ can be solved directly. For medium values of n, the problem cannot be solved easily by hand. A case is considered straightforward if the trivial upper bound UB_n can be calculated and a complete cover can be found for $L = UB_n$. In this situation, the optimal solution is known, i.e. $K = L = UB_n$. In other cases, however, K is smaller than UB_n, or a packing of size UB_n cannot be obtained easily. For such instances, a mathematical programming model is applied. The model takes the sizes of the squares as input, and a commercial solver is used. If the solver proves that there is no solution for a certain value L, but a solution exists for $L-1$, then the optimal solution is $K = L-1$.

3.5.1 Mathematical programming formulation

The covering problem is a special case of the Board Packing Problem (BoPP) [1, 12]. Two formulations were considered. The first is obtained by directly modifying the BoPP model, while the second is a specialized variant that exploits structural properties of the covering problem to reduce the number of constraints and variables.

3.5.2 Results using the mathematical models

The models were solved using the commercial mixed-integer programming solver CPLEX. For values in the range $n = 8, \ldots, 21$, the performance of the two models did not differ significantly, although the specialized model typically exhibited slightly lower running times when proving optimality. Table 1 summarizes the optimal results for $8 \leq n \leq 21$.

Any case with $UB_n = K$ is straightforward, since it is sufficient to find a good cover for UB_n. However, such a cover may not be easy to construct for relatively large n, for example $n = 21$. These are the cases $n \in \{9, 13, 14, 16, 18, 19, 20, 21\}$.

For the remaining cases, i.e. $n \in \{8, 10, 11, 12, 15, 17\}$, the optimal value of L can be determined by observing that CPLEX confirms the non-existence of a complete cover for $L+1$. The running time of CPLEX was approximately 7 seconds for $n = 11$ (to exclude $K = 22$ and confirm $K = 21$ as optimal). For $n = 12$, excluding $K = 25$ required about 12 seconds. For $n = 15$, excluding $K = 35$ required about 214 seconds. For $n = 17$, more than 10 hours of running time were needed to exclude $K = 42$, thus proving $K = 41$ as optimal. Consequently, for larger values of n, solving the models with CPLEX becomes computationally infeasible. Further details are provided in the thesis.

n	8	9	10	11	12	13	14	15	16	17	18	19	20	21
UB_n	14	16	19	22	25	28	31	35	38	42	45	49	53	57
K	13	16	18	21	24	28	31	34	38	41	45	49	53	57
Δ_K	2	3	2	3	3	4	3	3	4	3	4	4	4	4

Table 1: The cases $8 \leq n \leq 21$. The row Δ_K denotes the difference between consecutive K values as n increases.

3.6 Big cases for covering

Big cases are defined as those where CPLEX is far from being able to prove optimality. In such cases, CPLEX may provide some lower bounds, although this can already become time-consuming when the square to cover is larger than 70×70. Alternative approaches are therefore needed to generate lower bounds, including a heuristic algorithm and an expansion algorithm.

3.6.1 Heuristic search

For larger values of n, the commercial solver alone is insufficient. A heuristic can still be applied to find valid lower bounds on K for a given n. Specifically, the board packing problem [1] presented in Section 2.3 can be generated for given values of K and n, and a computer search can attempt to find a feasible solution with an objective function value that implies that a cover exists.

The heuristic used is an evolutionary algorithm, modified from the algorithm proposed in [1] to solve general instances of the board packing problem. The method is based on the hybrid genetic search, a leading metaheuristic successfully applied to highly competitive combinatorial optimization problems such as vehicle routing.

3.6.2 Expansion algorithm

Another approach for obtaining lower bounds is an *expansion algorithm*. Given an initial cover of a square of size $K \times K$ using consecutive squares up to size n, the expansion algorithm constructs a cover of a larger square $K' = K + s$ using consecutive squares of sizes from 2 to $n + 1$. The 1×1 square can then be added arbitrarily to complete the cover. The value of s, and thus K', depends on the minimum number of squares appearing in any row or column of the original cover.

The idea is more naturally expressed using rectangles. Consider a rectangle of size $K^I \times K^J$ covered by a set of n rectangles, where rectangle r has dimensions $I_r \times J_r$. The number of rectangles used in each row to cover the larger rectangle is counted. Let s denote the minimum number of such rectangles across all rows. A cover of a rectangle of size $K^I \times (K^J + s)$ can then be obtained by using n rectangles of dimensions $I_r \times (J_r + 1)$. This is achieved by extending each rectangle by one column and shifting them horizontally to cover the enlarged area.

Applying this procedure to a $K \times K$ square covered with n consecutive squares, the squares are first expanded horizontally. By transposing the result and applying the procedure again, a rectangle of size $(K + s) \times (K + s')$ covered by consecutive squares of sizes from 2 to $n+1$ is obtained. If $s \neq s'$, the process may be repeated while ensuring that the square is extended by equal length $\min\{s, s'\}$ in both directions.

Proposition 1. *The expansion algorithm provides a covering for each n and s.*

This procedure guarantees a cover, but not necessarily an optimal one.

3.6.3 Results using the heuristic and the expansion algorithm

For small values of K, the heuristic finds lower bounds comparable to those determined by the commercial MIP solver. Since the heuristic does not provide upper bounds, the MIP solver remains preferable up to values of K around 65. For larger K, the MIP solver is unable to provide conclusive results, and the heuristic is faster at producing lower bounds.

Table 2 reports results for $25 \leq n \leq 30$. The time required by the heuristic to find a cover for this range is typically less than one hour. However, when the heuristic fails to find a cover, it is not possible to determine whether the cover does not exist or whether the method simply failed to identify it. For $n = 25$, which was the largest case considered by CPLEX, the heuristic identified the same lower bound. The table also shows lower bounds obtained by applying the expansion algorithm. These were produced by taking the cover found by the heuristic for n and creating a new cover using $n+1$ squares. In this way, the expansion algorithm achieved the same lower bounds as the heuristic for $n = 26$ and $n = 28$ by extending solutions for $n = 25$ and $n = 27$, respectively.

n	25	26	27	28	29	30
UB_n	74	78	83	87	92	97
L	72	76	81	85	90	95
L'	NA	76	80	85	89	94

Table 2: The cases $25 \leq n \leq 30$ and results using the heuristic (L) and the expansion algorithm (L').

3.7 The asymptotic case

Recall that $N = \sqrt{A_n}$ with $A_n = \frac{n(n+1)(2n+1)}{6}$ is an upper bound on the side length of a square coverable by the squares $1 \times 1, 2 \times 2, \ldots, n \times n$. This bound is asymptotically tight, i.e., a cover can be constructed for a square of area at least $(1 - o(1))A_n$ as $n \to \infty$. More explicitly, a square of side length $N - 2n$ admits a cover with these squares. This fact is proved in the following more general form.

Theorem 3. *If $p \cdot q \leq A_n$ and $|p - q| < 3n$, then a $(p - 2n) \times (q - 2n)$ rectangle can be covered with the squares of side lengths $1, 2, \ldots, n$.*

4 New scientific results

This thesis contributes new scientific results in the field of combinatorial optimization through the introduction of the Board Packing Problem (BoPP), a novel hybrid formulation combining principles from facility location and two-dimensional bin packing. While traditional models treat these domains separately, BoPP integrates spatial gain functions with variable rectangle selection under cost constraints. Additionally, new lower and upper bounds for square packing and covering are introduced, improving upon existing results for small and medium instances. The developed models, algorithms, and benchmark sets provide both theoretical insight and practical tools for geometric facility deployment problems.

For an overview of the author's academic publications and related bibliographic data, see the MTMT profile:
`https://m2.mtmt.hu/gui2/?type=authors&mode=browse&sel=10071311`

Thesis 1. *A new optimization problem, called the Board Packing Problem (BoPP), has been defined on a board with m rows and n columns, where each cell holds an integer revenue. A set R of rectangles is available, each defined by integer height, width, and purchase cost. The goal is to select and place rectangles on the board, aligned with the board's sides, to maximize profit, calculated as the total revenue from covered cells minus the total rectangle cost. Rectangles may overlap, but revenue from any cell can be collected only once. Its investigation has been initiated.*

1.1. It is proved in different ways that the BoPP is $\mathcal{NP}$-hard in different settings.

1.2. Benchmark classes were constructed, each comprising multiple generated instances. To facilitate result interpretation, a graphical interface was developed. Furthermore, the problem was formally modeled as a mixed-integer linear program (MILP).

1.3. A specialized hybrid evolutionary algorithm was developed for the BoPP. It uses a constructive heuristic to generate the initial population, providing near-optimal solutions already at the start of the evolutionary process.

1.4. The instances were solved using the commercial solver CPLEX and a custom evolutionary algorithm. It was conducted a comprehensive evaluation of the results across six different benchmark instance sets.

The corresponding publications are: [1, 2, 12].

Thesis 2. *I addressed the following problem: given n squares with side lengths $[1, n]$, the task is to pack them, without rotation or overlap, into the smallest possible enclosing square. This problem is listed as sequence A005842 in the On-Line Encyclopedia of Integer Sequences (OEIS). I proposed new lower and upper bounds for the minimal enclosing square, and developed a simple algorithm suitable for large values of n.*

2.1. A new lower bound, LB_2, was introduced. For small and medium values of n (up to $n = 56$), the lower bounds were compared with known upper bounds from the literature. For larger values of n, a simple greedy-type algorithm was defined, and shown to be asymptotically optimal.

The corresponding publications are: [3, 11].

Thesis 3. *The following new problem was introduced: given n squares with side lengths* $[1, n]$*, the objective is to cover completely a square of size K using these items completely the largest square of size K. An evolutionary algorithm was developed, along with an expansion-based approach. These two algorithms, together with results obtained from CPLEX, were used to establish new lower-bound estimates for the largest possible value of K for various values of n.*

3.1. For small values of n, the optimal value of K was determined using a combination of simple observations, dedicated algorithms, and the commercial solver CPLEX.

3.2. For moderate values of n, lower and upper bounds for K were established through the use of CPLEX, an evolutionary heuristic, and an expansion algorithm.

3.3. For large values of n, a simple covering algorithm was developed that is proven to be asymptotically optimal.

The corresponding publications are: [4, 5].

5 Bibliography

[1] Abraham Gy., Dosa Gy., Hvattum L. M., **Olaj, T. A.**, Tuza Zs.: The board packing problem. *European Journal of Operational Research*, Volume 308, Issue 3, pages 1056–1073, `https://doi.org/10.1016/j.ejor.2023.01.030`, **D1 journal, IF=6.365**, 2023.

[2] Abraham, Gy., Dósa, Gy., Hvattum, L. M., **Olaj, T. A.**, Tuza, Zs.: Egy téglalap pakolási feladat, és megoldása különféle módszerekkel. *XXXV. Magyar Operációkutatási Konferencia*, Budapesti Corvinus Egyetem, Budapest, június 21.-23, 2023.

[3] Balogh, J., Dosa, Gy., Hvattum, L.M., **Olaj, T. A.**, Tuza, Zs.: Guillotine cutting is asymptotically optimal for packing consecutive squares. *Optimization Letters*, 16, pages 2775–2785, doi: 10.1007/s11590-022-01858-w, **Q1 journal, IF=1.502**, 2022.

[4] Balogh, J., Dosa Gy., Hvattum, L. M., **Olaj, T. A.**, Tuza, Zs.: Covering a square with consecutive squares. *MATCOS 2022 conference*, Koper, Slovenia, 2022.

[5] Balogh, J., Dósa, Gy., Hvattum, L. M., **Olaj, T. A.**, Szalkai, I., Tuza, Zs.: Covering a square with consecutive squares. *Annals of Operational Research, Springer*, https://doi.org/10.1007/s10479-025-06633-5, **Q1 journal, IF=4.82**, 2025.

[6] Bereg, S., Cabello, S., Diaz-Banez, J. M., Pérez-Lantero, P., Seara, C., Ventura, I.: The class cover problem with boxes. *Computational Geometry*, 45(7), pages 294–304, 2012.

[7] Berman, O., Krass, D.: The generalized maximal covering location problem. *Computers & Operations Research*, 29, pages 563–581, 2002.

[8] Blanquero R., Carrizosa E., G.-Tóth B.: Maximal Covering Location Problems on networks with regional demand. *Omega*, 64(C), pages 77-85, 2016.

[9] Caprara, A., Monaci, M.: On the two-dimensional Knapsack Problem. *Operations Research Letters*, Volume 32, Issue 1, pages 5–14, 2004.

[10] Demiröz, B.E., Altınel, K., Akarun, L.: Rectangle blanket problem: Binary integer linear programming formulation and solution algorithms. *European Journal of Operational Research*, 277, pages 62–83, 2019.

[11] Dosa, Gy., **Olaj, T. A.**: On Square Packing. *Veszprem Optimization Workshop (VOW)*, Veszprem, 2019.

[12] Dosa, Gy., Hvattum, L. M., **Olaj, T. A.**, Tuza, Zs.: The board packing problem: Packing rectangles into a board to maximize profit. In I. Vassányi, editor, *Proceedings of the Pannonian Conference on Advances in Information Technology (PCIT 2020)*, pages 10–16. University of Pannonia, Veszprém, Hungary, 2020.

[13] Gardner, M.: Mrs. Perkins' quilt and other square-packing problems. *In Mathematical Carnival, New York: Alfred A. Knopf*, pages 139–149, 1975.

[14] Ghaffarian, S., Kerle, N.: Towards post-disaster debris identification for precise damage and recovery assessments from UAV and satellite images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13, pages 297–302, 2019.

[15] Guo, D., Li, X., Lou, W.: Detection and analysis of road information based on optical satellite image. *2021 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pages 94–96, 2021.

[16] Hougardy, S.: A scale invariant algorithm for packing rectangles perfectly. *Proceedings of the fourth International Workshop on Bin Packing and Placement Constraints (BPPC'12)*, http://www.or.uni-bonn.de/~hougardy/paper/PerfectPacking.pdf, 2012.

[17] Jang, J., Choi, J., Bae, H.-J., Choi, I.-C.: Image collection planning for KOrea Multi-Purpose SATellite-2. *European Journal of Operational Research*, 230, pages 190–199, 2013.

[18] Leung, J. Y.-T., Tam, T. W., Wong, C. S., Young, G. H., Chin, F. Y. L.: Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10(3), pages 271–275, 1990.

[19] Lodi, A., Martello, S., Monaci, M.: Two-dimensional packing problems: A survey. *European Journal of Operational Research*, Volume 141, Issue 2, pages 241–252, 2002.

[20] Masek, W. J.: Some *NP*-complete set covering problems. *Unpublished manuscript*, 1978.

[21] Mondino, E. B., Perotti, L., Piras, M.: High resolution satellite images for archeological applications: the Karima case study (Nubia region, Sudan). *European Journal of Remote Sensing*, 45(1), pages 243–259, 2012.

[22] Scheithauer, G.: Introduction to Cutting and Packing Optimization: Problems, Modeling Approaches. *Solution Methods (International Series in Operations Research & Management Science*, 263, Springer, 2018.

[23] Sgall J., Balogh J., Békési J., Dósa Gy., Hvattum L. M., Tuza Zs.: No Tiling of the 70*70 Square with Consecutive Squares. *In 12th International Conference on Fun with Algorithms (FUN 2024). Leibniz International Proceedings in Informatics (LIPIcs)*, volume 291, pages 28:1–28:16, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, `https://doi.org/10.4230/LIPIcs.FUN.2024.28`, 2024.

[24] The on-line encyclopedia of integer sequences, sequence a005842, `http://oeis.org`, Accessed: March 26, 2021.