Maximum Polygon Packing: The CG Challenge 2024

Sándor P. Fekete
Department of Computer Science, TU Braunschweig, Germany

Phillip Keldenich
Department of Computer Science, TU Braunschweig, Germany

Dominik Krupke
Department of Computer Science, TU Braunschweig, Germany

Stefan Schirra
Department for Simulation and Graphics, OvGU Magdeburg, Germany

Abstract

We give an overview of the 2024 Computational Geometry Challenge targeting the problem Maximum Polygon Packing: Given a convex region P in the plane, and a collection of simple polygons $Q_1, \ldots, Q_n$, each Q_i with a respective value c_i, find a subset $S \subseteq \{1, \ldots, n\}$ and a feasible packing within P of the polygons Q_i (without rotation) for $i \in S$, maximizing $\sum_{i \in S} c_i$. Geometric packing problems, such as this, present significant computational challenges and are of substantial practical importance.

Keywords and phrases Computational Geometry, geometric optimization, packing, Algorithm Engineering, contest

Digital Object Identifier 10.57717/cgt.v5i5.125

Funding Work at TU Braunschweig has been partially supported by the German Research Foundation (DFG), project "Computational Geometry: Solving Hard Optimization Problems" (CG:SHOP), grant FE407/21-1 and FE407/21-2.

Acknowledgements We thank the members of the CG:SHOP Challenge Advisory Board for their valuable input: William J. Cook, Andreas Fabri, Dan Halperin, Michael Kerber, Philipp Kindermann, Joe Mitchell, Kevin Verbeek.

1 Introduction

The "CG:SHOP Challenge" (Computational Geometry: Solving Hard Optimization Problems) originated as a workshop at the 2019 Computational Geometry Week (CG Week) in Portland, Oregon in June, 2019. The goal was to conduct a computational challenge competition that focused attention on a specific hard geometric optimization problem, encouraging researchers to devise and implement solution methods that could be compared scientifically based on how well they performed on a database of carefully selected and varied instances. While much of computational geometry research is theoretical, often seeking provable approximation algorithms for NP-hard optimization problems, the goal of the Challenge was to set the metric of success based on computational results on a specific set of benchmark geometric instances. The 2019 Challenge [25] focused on the problem of computing simple polygons of minimum and maximum area for given sets of vertices in the plane. It generated a strong response from many research groups [11, 37, 21, 24, 57, 44] from both the computational geometry and the combinatorial optimization communities, and resulted in a lively exchange of solution ideas.

Subsequently, the CG:SHOP Challenge became an event within the CG Week program, with top performing solutions reported in the Symposium on Computational Geometry (SoCG) proceedings. The schedule for the Challenge was advanced earlier, to give an opportunity for more participation, particularly among students, e.g., as part of course projects. For CG Weeks 2020, 2021, 2022, and 2023, the Challenge problems were Minimum Convex Partition [18, 66, 55, 20], Coordinated Motion Planning [26, 14, 65, 13, 46, 64], Minimum Partition into Plane Subgraphs [27, 10, 59, 12, 60, 35, 58], and Minimum Convex Covering [28, 15, 1], respectively.

The sixth edition of the Challenge in 2024 continued this format, leading to contributions in the SoCG proceedings.

2 The challenge: maximum polygon packing

A suitable contest problem has a number of desirable properties.

- The problem is of geometric nature.
- The problem is of general scientific interest and has received previous attention.
- Optimization problems tend to be more suitable than feasibility problems; in principle, feasibility problems are also possible, but they need to be suitable for sufficiently fine-grained scoring to produce an interesting contest.
- Computing optimal solutions is difficult for instances of reasonable size.
- This difficulty is of a fundamental algorithmic nature, and not only due to issues of encoding or access to sophisticated software or hardware.
- Verifying feasibility of provided solutions is relatively easy.

In this sixth year, a call for suitable problems was communicated in March 2023. In response, a number of interesting problems were proposed for the 2024 Challenge. These were evaluated with respect to difficulty, distinctiveness from previous years, and existing literature and related work. In the end, the Advisory Board selected the chosen problem. Special thanks go to Mikkel Abrahamsen (University of Copenhagen) who suggested this problem, motivated by a rich history in geometry and optimization, including [45] and a wide range of previous work described further down.

2.1 The problem

The specific problem that formed the basis of the 2024 CG Challenge was the following.

Problem: Maximum Polygon Packing
Given: A convex region P in the plane, and a collection of simple polygons $Q_1, \dots, Q_n$, each Q_i with a respective value c_i.
Goal: Find a subset $S \subseteq \{1, \dots, n\}$ and a feasible packing within P of the polygons Q_i (without rotation) for $i \in S$, maximizing $\sum_{i \in S} c_i$.

2.2 Related work

Problems of geometric packing have been studied for a long time, with classic challenges such as the Kepler conjecture [41, 39, 38] on densest sphere packings. Providing a survey that does justice to the wide range of relevant work goes beyond the scope of this overview; we refer to Fejes Tóth [23, 63], Lodi, Martello and Monaci [47], Brass, Moser and Pach [8] and Böröczky [7] for more comprehensive surveys on the theoretical side, and Dyckhoff and Finke [19], Sweeney and Paternoster [61], Bennell et al. [6, 5], Leao et al. [43] on the practical side.

Even the decision problem whether it is possible to pack a given set of (not necessarily identical) axis-aligned squares into the unit square was shown to be strongly NP-complete by

Leung et al. [45], using a reduction from 3-Partition. Additional difficulties arise when rotation of the squares is allowed, even when packing unit squares into a square [22, 36, 9]. For packing polygons, Allen and Iacono [3] showed that even packing a maximum number of *identical* simple polygons into a square is NP-complete. For packing a set of nonidentical polygons, Abrahamsen et al. [2] gave a general framework for establishing ∃ℝ-hardness, making it unlikely that the problem belongs to NP.

Despite these theoretical difficulties, there is also a wide range of positive results. Already in 1967, Moon and Moser [56] proved that it is possible to pack a set of squares into the unit square if their total area does not exceed $1/2$, which is best possible. More recently, Merino and Wiese [49] considered two-dimensional Knapsack Problems for weighted convex polygons, which is closely related to our Challenge problem. They gave a number of approximation algorithms, even in the presence of rotation. From the context of practical Computational Geometry, Milenkovic et al. [17, 54, 50, 53, 52, 51] provided a spectrum of heuristic methods, while Fekete and Schepers [34, 33, 29, 30, 31, 32] developed exact methods for optimally packing axis-aligned orthogonal objects in two and higher dimensions.

2.3 Instances

Creating suitable instances is a critical aspect of any challenge. Instances that are too easy to solve undermine the challenge's complexity, making it trivial. Conversely, instances that demand extensive computational resources for basic preprocessing or for finding viable solutions can provide an unfair advantage to teams with superior computing facilities. This issue is exacerbated when the instance set is too voluminous to be managed by a small number of computers.

We utilized four distinct instance generators for the contest, each permitting the adjustment of various parameters, such as item complexity or variance. This approach ensures a diverse collection of instances. The generators are described as follows.

random Instances are created by generating a set of random polygons and a random container. The process involves generating several random points and then forming a polygon from them: starting from their convex hull, each remaining point is admitted to the boundary with a certain probability (determined by a random ratio), yielding polygons ranging from convex to increasingly concave, enabling the production of instances with items whose sizes and complexities adhere to configurable distributions. See Figure 1 for an example.

jigsaw These instances originate from segmenting the container or its bounding box into pieces, then slightly altering the pieces to hinder easy reassembly. The segmentation is performed by first adding a number of random lines through the container or its bounding box. This results only in convex and often relatively simple pieces, such that in a second step, we go through the geometric arrangement and randomly combine adjacent faces to form more complex pieces. During this process, we make sure not to create non-simple polygons or generally badly shaped or sized pieces. Given the nature of the Knapsack Problem, an item surplus is generated by amalgamating multiple jigsaws, thereby forming a complex instance. See Figure 2 for an example.

atris The **atris** generator uses a rectangular strip of some specified width and height as the container. It generates items that are derived from polyominos in the well-known computer game Tetris. More specifically, the generator creates items from seven different categories: *Line*, *Squiggly*, *Double Squiggly*, *Y*, *T*, *L* and *Plus*.
However, unlike in Tetris, the width and height of the pixels constituting these items are not fixed, but individual columns and rows of pixels have their width and height chosen

uniformly at random between some minimum and maximum value. Furthermore, for items consisting of multiple arms such as a *Y*, a *Plus* or a *Squiggly*, each of the arms can be shifted by some randomly chosen distance, such that the resulting shape need not be symmetric. This shifting is restricted to prevent disconnecting the item or change it to another type of shape.

Each item can be flipped vertically and horizontally based on a fair coin toss; furthermore, each item is rotated by an integer multiple of 90°, chosen between 0 and 3 uniformly at random.

Item generation stops once the total item area exceeds some preselected multiple $t \in [1, 2]$ of the container area; instances with a larger number of items use larger containers.

The value of each item is derived by multiplying a random scaling factor from $[0.8, 1.2]$ with the item's area, further multiplying the result with a constant depending on the item's shape category, giving more weight to items that are arguably harder to pack (such as Y-shapes and Double Squigglies) and less weight to easier shapes such as rectangles. The resulting value is then rounded to an integer.

We took care to avoid any possible overflow or imprecision when working with the scores; all scores and coordinates generated by this generator are integral and even for the largest contest instances, the sum of all item scores was below 2^{40}, well below the threshold at which `double` values are no longer able to precisely encode integers.

See Figure 3 for an example.

satris The **satris** generator works nearly identically to the **atris** generator; in addition, items are randomly selected for an additional shearing transformation, i.e., they have their coordinates transformed according to a matrix $\left(\begin{smallmatrix} 1 & m \\ 0 & 1 \end{smallmatrix}\right)$, with $m \in [0.1, 2]$ chosen uniformly at random; this transformation happens before the flipping and rotating transformation outlined above.

Sheared items have all their coordinates rounded to the nearest integer. Their value is increased compared to non-sheared items, both by multiplication with a fixed constant and by multiplication with another constant increasing with m, so items that are more severely sheared receive higher value.

See Figure 4 for an example.

In addition to the geometric shapes of the polygons, the value attributed to each polygon plays a crucial role in the problem instances. To construct our instances, we employed several value functions, outlined as follows.

Area: The value of a polygon is directly proportional to its area.
Convex Hull Area: A polygon's value corresponds to the area of its convex hull.
Rotated Bounding Box: The value is based on the area of the polygon's smallest enclosing rectangle.
Uniform: Each polygon is assigned a uniform value, regardless of its size.

To introduce further variability into our instances, we incorporated noise into the value functions for certain cases. To this end, the value is adjusted by multiplying it with a minor random factor, encouraging a strategic approach to item prioritization beyond mere density optimization. The specific value functions utilized were not revealed to the challenge participants.

Using the described generators and value functions, we created a large set of instances, of which a subset was selected for the contest. This subset was selected by first identifying a set of eleven metrics that describe the instances and allow quantifying their differences. These metrics included the logarithm of the number of items, the difference of the area to

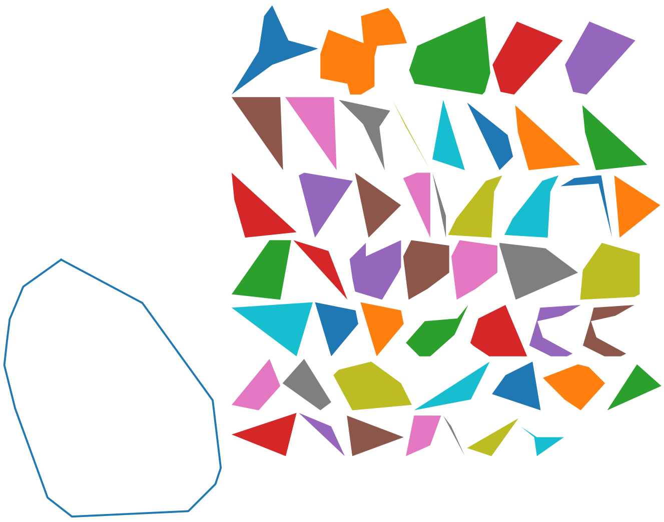


Figure 1 An example of a **random** instance.

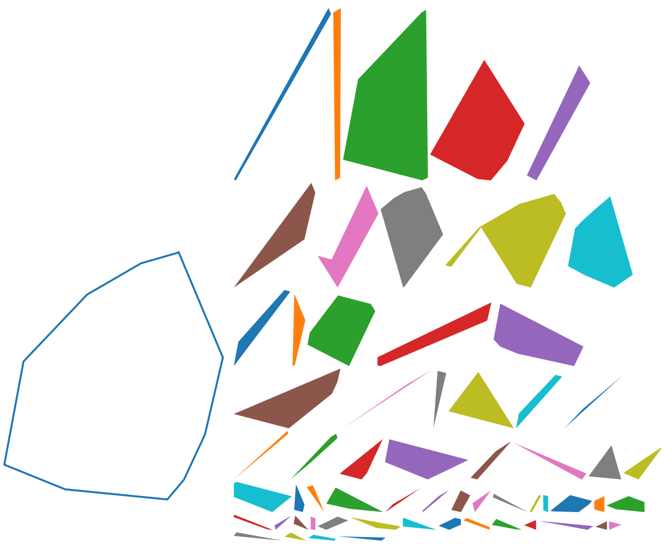


Figure 2 An example of a **jigsaw** instance.

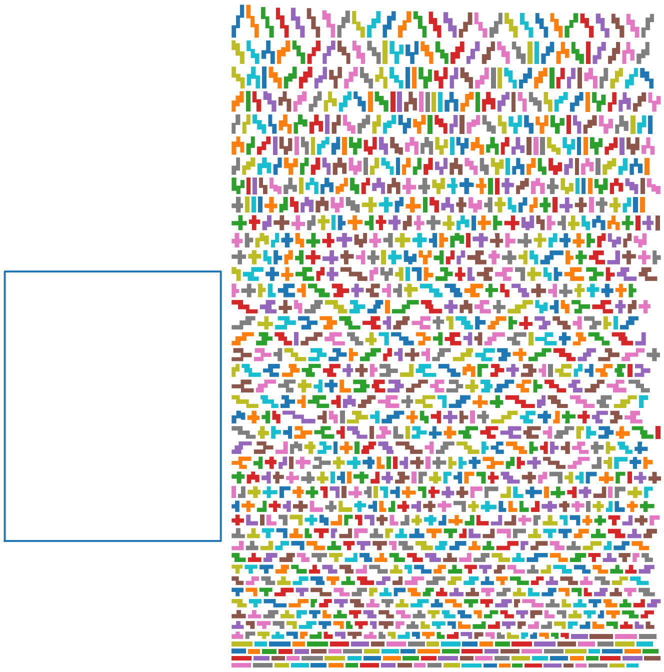


Figure 3 An example of an **atris** instance.

the area of the convex hull, the difference to a (rotated) rectangle of container and items, the relation of area of items to the container area, a rough measure on how aligned the items are, and further. As these metrics have some correlations and just computing the geometric differences would put too much weight on specific aspects even after normalization, we used a principal component analysis (PCA) to reduce the dimensionality, and thus correlation, of the instance space. To select a diverse set of instances of a specific size, we then grouped similar instances into the desired number of groups using a k-means clustering algorithm, and selected a random instance from each group for the final benchmark. The instance distribution of the initial and final set of instances can be seen in Figure 5. A more detailed description of this approach can be found in Chapter 10 of [42].

2.4 Evaluation

The competition featured a total of 180 instances. For each instance I, define $B(I)$ as the highest value achieved by any team's best solution for I. Similarly, let $T(I)$ represent the value of team T's best solution for I. The score $S_T(I)$ for team T on instance I is calculated as:

$$S_T(I) := \frac{T(I)^2}{B(I)^2}.$$

Thus, if a team's value is half that of the best solution, their score for this instance reduces to 0.25. Teams achieving the highest submitted value for I receive a perfect score of 1 for that instance. Teams failing to submit any valid solution for I are assigned a score of 0, corresponding to a solution that fails to pack any items into the container. The overall score S_T for each team T is the sum of $S_T(I)$ across all instances. The contest winner is the team

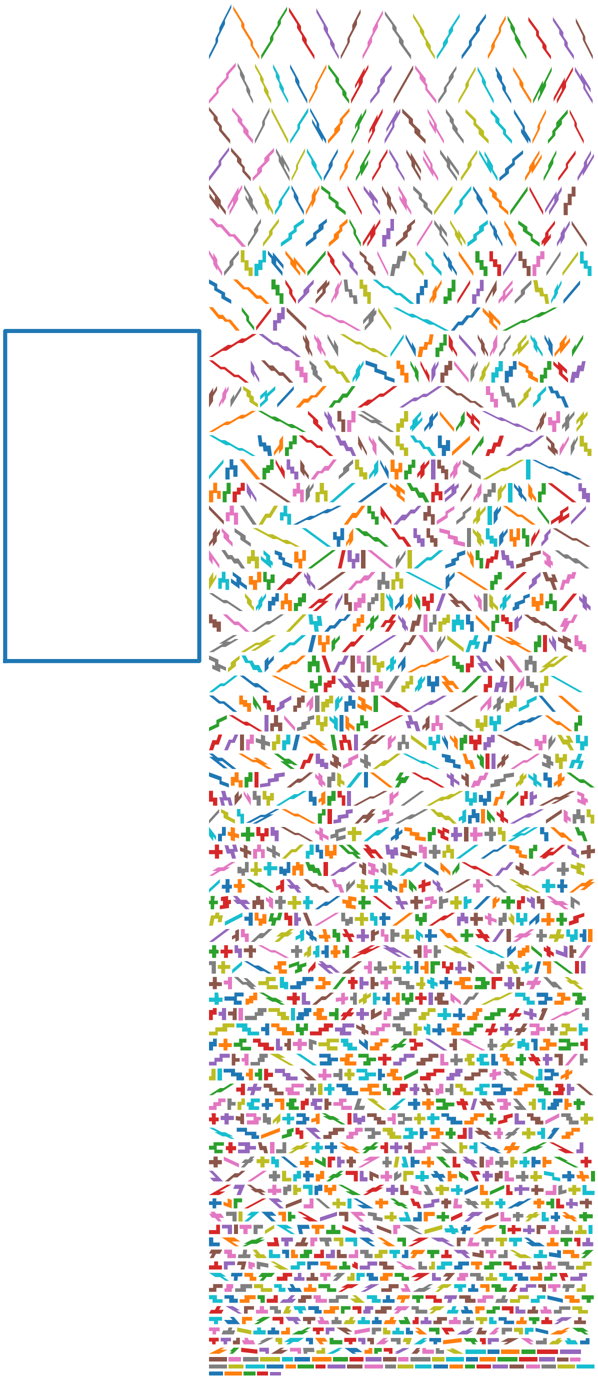
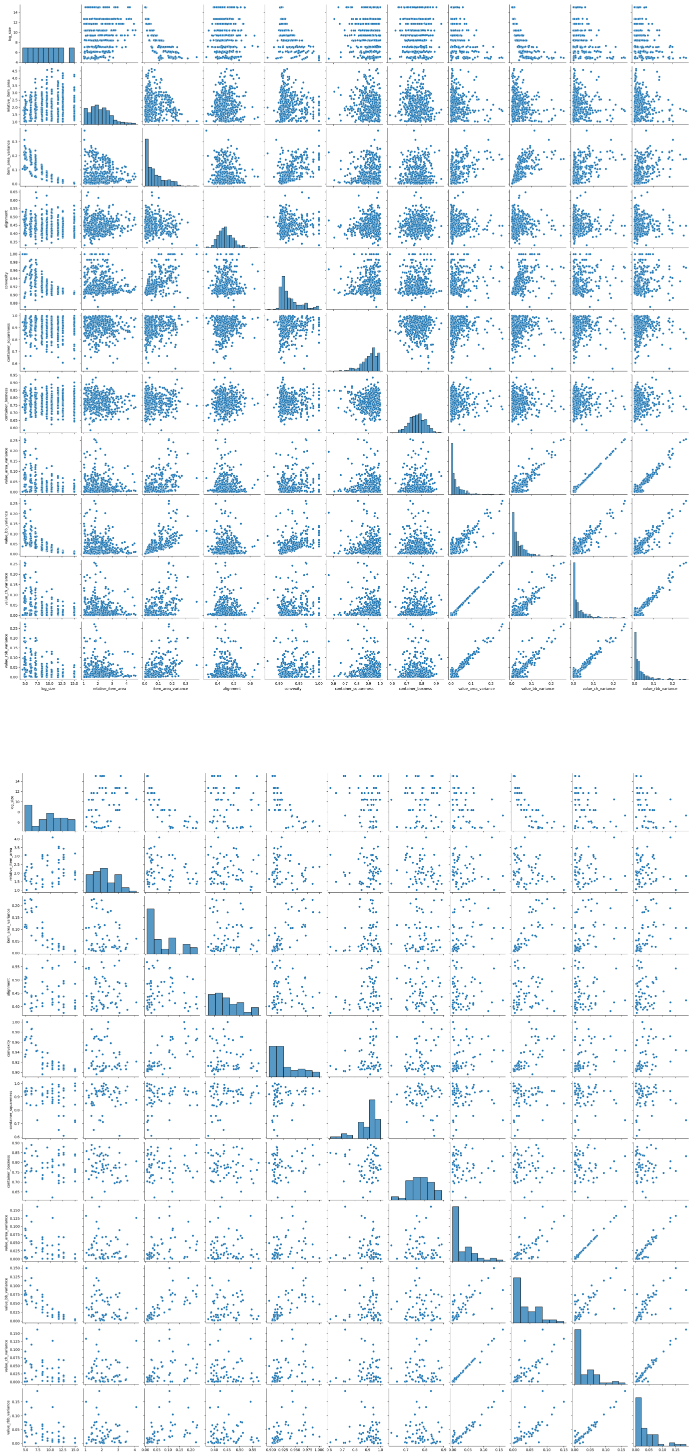


Figure 4 An example of a **satris** instance.



■ **Figure 5** Pair plots of the instance metrics for the initial set of instances (top) and the final benchmark (bottom), showing the correlation between the different metrics and the distribution of the instances among them.

with the highest total score. In the event of a tie, the deciding factor would have been the time at which the tied score was first achieved; just like in the past, this tiebreaker was not needed.

The benefit of employing a relative scoring system lies in its independence from the quality of bounds or approximations, instead directly comparing team performances. While scoring relative to a lower or upper bound offers consistency, the impact of an instance on the overall score may significantly depend on the bound's accuracy. This approach could inadvertently diminish the influence of challenging and engaging instances—often characterized by less precise bounds—on the total score.

2.5 Categories

The contest was run in an *Open Class*, in which participants could use any computing device, any amount of computing time (within the duration of the contest) and any team composition.

2.6 Server and timeline

The competition was facilitated through a dedicated server at TU Braunschweig, accessible at `https://cgshop.ibr.cs.tu-bs.de/competition/cg-shop-2024/`. An initial batch of example instances was made available on July 31, 2023, followed by the release of the final benchmark set on September 29, 2023. The competition concluded on January 22, 2024 (AoE).

Participants were provided with a verification tool as an open-source Python package, available at `https://github.com/CG-SHOP/pyutils24`. This tool provided detailed information on errors and allowed participants to easily investigate and correct their solutions if the server rejected them. To ensure accurate verification and address potential floating point arithmetic issues, the tool leveraged the CGAL library [62]. Given that many instances feature items of similar sizes, leading to a homogeneous distribution in optimal packings, the adoption of a simple quad tree structure significantly accelerated the verification process. This approach enabled the selective examination of nearby items for intersections, facilitating nearly instantaneous solution verification, even in the presence of the computational demands of exact arithmetic.

3 Outcomes

Table 1 outlines the outcomes for the participating teams, ranked by their performance. Overall, nine teams participated and agreed to be evaluated and ranked. Four teams reached scores above 150 of a possible 180 and were invited for contributions in the 2024 SoCG proceedings, as follows.

1. Team Shadoks: Guilherme Dias da Fonseca and Yan Gerard, "Shadoks Approach to Knapsack Polygonal Packing" [16].
2. Team SmartPlacer: Canhui Luo, Zhouxing Su and Zhipeng Lü, "A General Heuristic Approach for Maximum Polygon Packing" [48].
3. Team CGA Lab Salzburg: Martin Held, "Priority-Driven Nesting of Irregular Polygonal Shapes Within a Convex Polygonal Container Based on a Hierarchical Integer Grid" [40].
4. Team TU Dortmund: Alkan Atak, Kevin Buchin, Mart Hagedoorn, Jona Heinrichs, Karsten Hogreve, Guangping Li, Patrick Pawelczyk, "Computing Maximum Polygonal Packings in Convex Polygons using Best-Fit, Genetic Algorithms and ILPs" [4].

Team Shadoks used Integer Linear Programming (ILP) and a carefully designed greedy approach for initial solutions, which were then improved by local search. SmartPlacer

Rank	Team	Score	# best	# only ≥ 0.95	# only ≥ 0.9
1	Shadoks	173.91	71	32	12
2	SmartPlacer	167.67	58	28	2
3	CGA Lab Salzburg	156.62	9	3	0
4	TU Dortmund	154.91	37	19	8
5	AMW	146.64	1	0	0
6	GreedyPackers	128.63	0	0	0
7	polygonizers	89.06	0	0	0
8	QTeam	72.59	0	0	0
9	CodingAtChristmas	3.39	0	0	0

Table 1 Competition scores of the teams that submitted at least one valid solution and opted to be evaluated and ranked. The columns # best, # only ≥ 0.95, and # only ≥ 0.9 show the number of instances for which the team achieved the best score, as only team a score of at least 0.95, and as only team a score of at least 0.9, respectively.

subdivided large problems into smaller subproblems, for which they tried to fill a subcontainer sequentially, using a combination of local and tabu search to arrive at conflict-free placements. CGA Lab Salzburg used a hierarchical grid and priority heuristics for filling the container polygon sequentially. TU Dortmund's approach involved different strategies depending on problem size: ILPs were used for small, genetic algorithms for medium and a best-fit approach for large instances.

The top-ranking team, Shadoks, achieved an outstanding score of 173.91, leading the competition. Close behind, SmartPlacer scored 167.67. CGA Lab Salzburg and TU Dortmund follow with scores of 156.62 and 154.91, respectively. AMW and GreedyPackers also demonstrated commendable efforts with scores just above 146 and 128, respectively. The progress over time of each team's score can be seen in Figure 6; overviews of best solutions are shown in the subsequent figures.

The competition data reveals insightful trends in team performance across various metrics, as shown in the accompanying figures. Figure 6 illustrates the dynamic evolution of solution quality, highlighting continuous improvements by team TU Dortmund and a strategic late entry by CGA Lab Salzburg, with a surge in activity in the final weeks. Figure 7 indicates a significant increase in submissions as the contest neared its end, albeit with a less pronounced last-day spike compared to previous competitions. Performance analysis across different instance types in Figure 8 shows team Shadoks excelling in **satris** instances, TU Dortmund in **atris**, and SmartPlacers demonstrating strength in **jigsaw** and **random** instances, with CGA Lab Salzburg and AMW maintaining balanced performances. Figure 9 analyzes performance trends relative to instance size, with SmartPlacer dominating in smaller instances, Shadoks leading in larger ones, and TU Dortmund displaying a notable rebound in performance for the largest instances. Finally, Figure 10 assesses the impact of value functions on team scores, highlighting the challenges faced by TU Dortmund with noise-infused values, the proficiency of SmartPlacer with uniform values, and the dominance of Shadoks in area-based value functions. Collectively, these figures highlight the complexities and strategic nuances of the competition, reflecting the diverse strengths and adaptabilities of the participating teams, and hint at the perspectives for further improvements.

Note that this analysis may be influenced by correlations due to the uneven distribution of metrics. For instance, atris and satris instances exclusively use area as their value function, whereas random and jigsaw instances incorporate all four value functions. This discrepancy

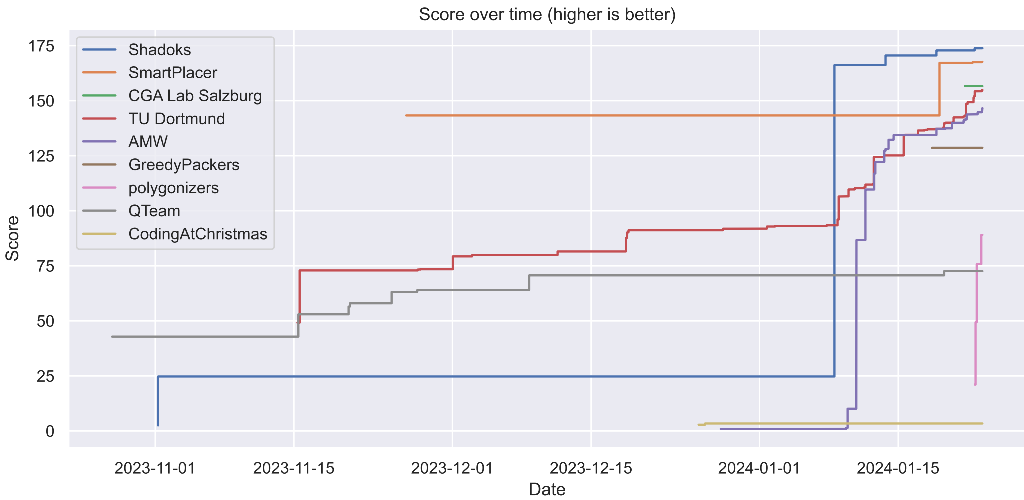


Figure 6 The plot traces the evolution of solution quality throughout the contest. The overall score, derived from the sum of scores for individual instances based on final submissions, illustrates a steady progression. Notably, team TU Dortmund exhibits incremental improvements, culminating in a score that more than doubles. CGA Lab Salzburg makes a late yet impactful entry. The final two weeks witness significant activity across teams.

could affect the interpretation of results and comparisons across different instance types. You can investigate the solutions of the top teams for individual instances on `https://pageperso.lis-lab.fr/guilherme.fonseca/cgshop24view/` (developed by the team Shadoks).

4 Conclusions

The 2024 CG:SHOP Challenge motivated a considerable number of teams to engage in extensive optimization studies. The outcomes promise further insight into the underlying, important optimization problem. This demonstrates the importance of considering geometric optimization problems from a practical perspective.

References

1. M. Abrahamsen, W. B. Meyling, and A. Nusser. Constructing concise convex covers via clique covers. In *Symposium on Computational Geometry (SoCG)*, volume 258 of *LIPIcs*, pages 66:1–66:9, 2023.
2. M. Abrahamsen, T. Miltzow, and N. Seiferth. Framework for er-completeness of two-dimensional packing problems. In *Symposium on Foundations of Computer Science (FOCS)*, pages 1014–1021, 2020.
3. S. R. Allen and J. Iacono. Packing identical simple polygons is NP-hard. *arXiv preprint arXiv:1209.5307*, 2012.
4. A. Atak, K. Buchin, M. Hagedoorn, J. Heinrichs, K. Hogreve, G. Li, and P. Pawelczyk. Computing maximum polygonal packings in convex polygons using best-fit, genetic algorithms and ilps. In *Symposium on Computational Geometry (SoCG)*, volume 293 of *LIPIcs*, pages 86:1–86:9, 2024.
5. J. A. Bennell and J. F. Oliveira. The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184(2):397–415, 2008.

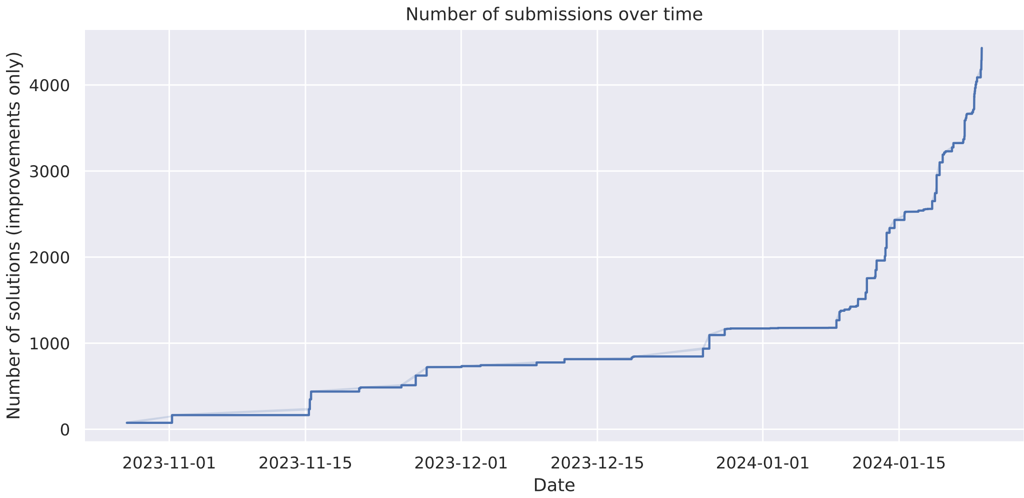


Figure 7 This plot depicts the number of submissions over time, highlighting a concentration of activity in the final two weeks. Unlike in previous Challenge iterations, the surge in submissions on the final day is less pronounced.

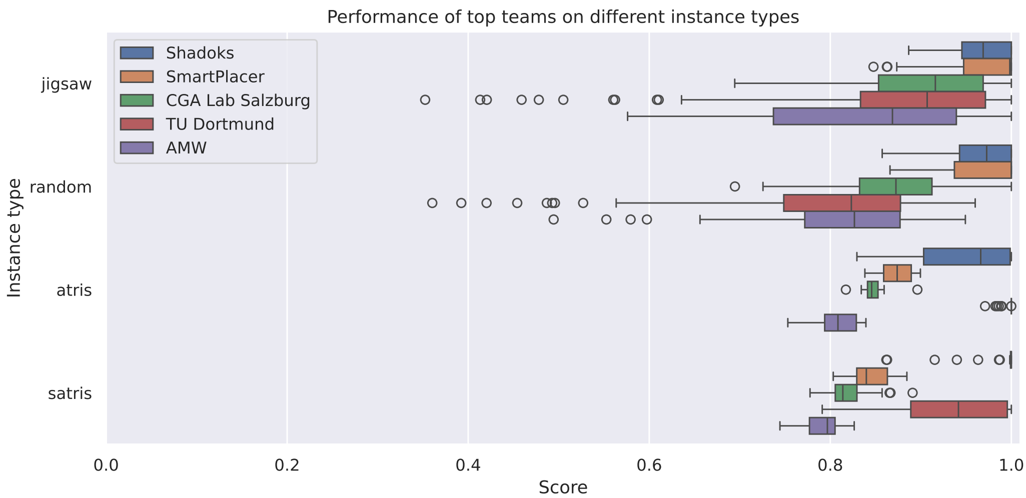


Figure 8 This plot contrasts the performance of the top five teams across different instance types, illustrating the distribution of scores. As the scores are relative to the best solution, this allows for a straightforward comparison of team performances. The box plot depicts the median, quartiles, and outliers. Here, an outlier is specified as a score lying beyond 1.5 times the interquartile range above the upper quartile or below the lower quartile, offering insights into the variability and central tendency of each team's scores. We can see that team Shadoks leads in **satris** instances, while TU Dortmund excels in **atris** instances. Team SmartPlacers shows strength in **jigsaw** and **random** instances, albeit with weaker performance elsewhere. CGA Lab Salzburg and AMW exhibit no particular preference for instance types.

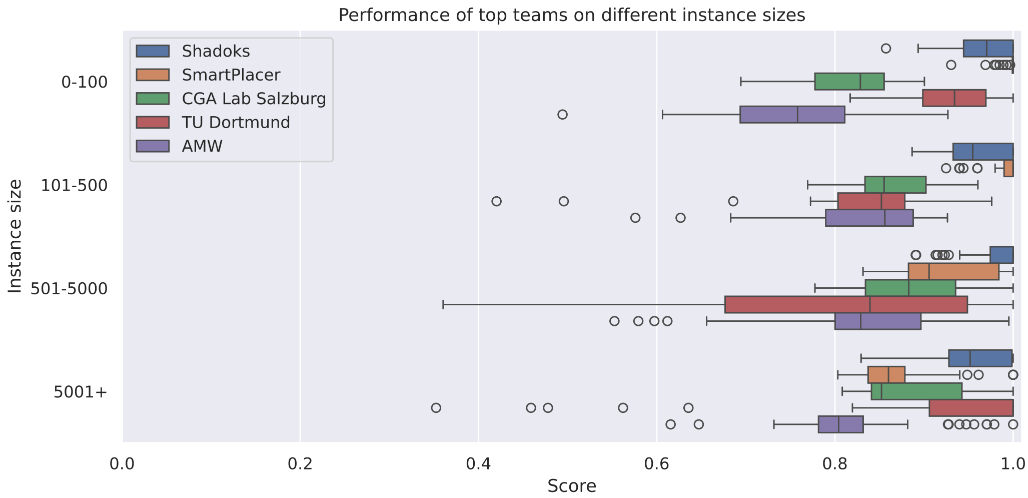


Figure 9 This plot examines how the top five teams fare across different instance sizes. SmartPlacer dominates instances with up to 500 items. The team Shadoks emerge as the front-runner for instances sized between 501 and 5000 items. No distinctive patterns of dominance are observed for CGA Salzburg and AMW. Interestingly, TU Dortmund shows a dip in performance for medium-sized instances, but rebounds strongly for larger instances.

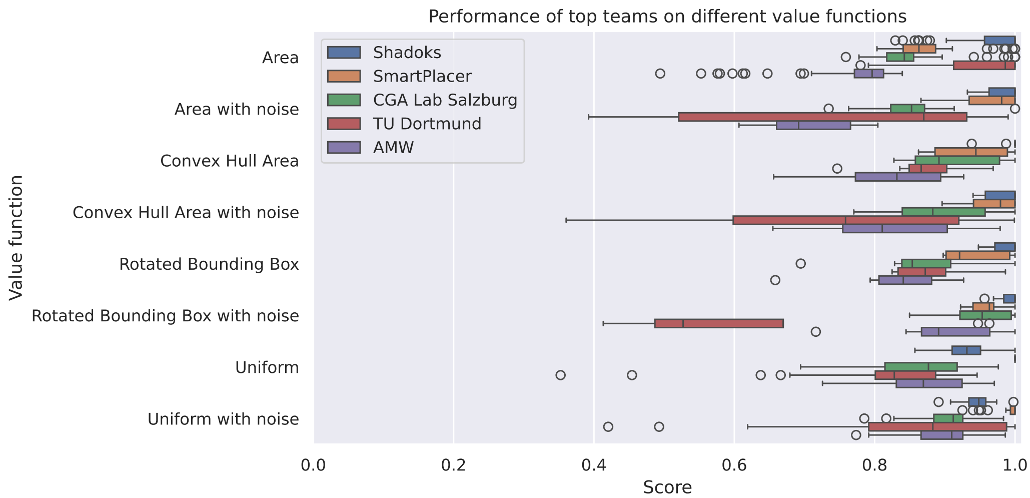


Figure 10 This plot assesses the impact of different value functions on the scores of the top five teams. TU Dortmund experiences a decline in performance with noise-infused value functions. Team Shadoks slightly falters in scenarios for which items have uniform values, a domain for which SmartPlacer excels. Shadoks, however, dominates in instances for which value functions are determined by the area of the convex hull.

6 J. A. Bennell and J. F. Oliveira. A tutorial in irregular shape packing problems. *Journal of the Operational Research Society*, 60:S93–S105, 2009.

7 K. Böröczky Jr. *Finite Packing and Covering (Cambridge Tracts in Mathematics)*. Cambridge University Press, 2004.

8 P. Brass, W. O. J. Moser, and J. Pach. *Research Problems in Discrete Geometry*. Springer Science & Business Media, 2005.

9 F. Chung and R. Graham. Efficient packings of unit squares in a large square. *Discrete & Computational Geometry*, 64(3):690–699, 2020.

10 L. Crombez, G. D. da Fonseca, F. Fontan, Y. Gerard, A. Gonzalez-Lorenzo, P. Lafourcade, L. Libralesso, B. Momège, J. Spalding-Jamieson, B. Zhang, and D. W. Zheng. Conflict optimization for binary CSP applied to minimum partition into plane subgraphs and graph coloring. *Journal of Experimental Algorithms*, 28:1.2:1–1.2:13, 2023.

11 L. Crombez, G. D. da Fonseca, and Y. Gerard. Greedy and local search solutions to the minimum and maximum area. *Journal of Experimental Algorithmics*, 27:2.2:1–2.2:11, 2022.

12 L. Crombez, G. D. da Fonseca, Y. Gerard, and A. Gonzalez-Lorenzo. Shadoks approach to minimum partition into plane subgraphs. In *Symposium on Computational Geometry (SoCG)*, volume 224 of *LIPIcs*, pages 71:1–71:8, 2022.

13 L. Crombez, G. D. da Fonseca, Y. Gerard, A. Gonzalez-Lorenzo, P. Lafourcade, and L. Libralesso. Shadoks approach to low-makespan coordinated motion planning. In *Symposium on Computational Geometry (SoCG)*, volume 189 of *LIPIcs*, pages 63:1–63:9, 2021.

14 L. Crombez, G. D. da Fonseca, Y. Gerard, A. Gonzalez-Lorenzo, P. Lafourcade, and L. Libralesso. Shadoks approach to low-makespan coordinated motion planning. *Journal of Experimental Algorithms*, 27:3.2:1–3.2:17, 2022.

15 G. D. da Fonseca. Shadoks approach to convex covering. In *Symposium on Computational Geometry (SoCG)*, volume 258 of *LIPIcs*, pages 67:1–67:9, 2023.

16 G. D. da Fonseca and Y. Gerard. Shadoks approach to knapsack polygonal packing. In *Symposium on Computational Geometry (SoCG)*, volume 293 of *LIPIcs*, pages 83:1–83:9, 2024.

17 K. Daniels and V. J. Milenkovic. Multiple translational containment, Part I: An approximate algorithm. *Algorithmica*, 19(1):148–182, 1997.

18 E. D. Demaine, S. P. Fekete, P. Keldenich, D. Krupke, and J. S. B. Mitchell. Computing convex partitions for point sets in the plane: The CG:SHOP Challenge 2020, 2020.

19 H. Dyckhoff and U. Finke. *Cutting and packing in production and distribution: A typology and bibliography*. Springer Science & Business Media, 1992.

20 G. Eder, M. Held, S. de Lorenzo, and P. Palfrader. Computing low-cost convex partitions for planar point sets based on tailored decompositions. In *Symposium on Computational Geometry (SoCG)*, volume 164 of *LIPIcs*, pages 85:1–85:11, 2020.

21 G. Eder, M. Held, S. Jasonarson, P. Mayer, and P. Palfrader. 2-opt moves and flips for area-optimal polygonalizations. *Journal of Experimental Algorithmics*, 27:2.7:1–2.7:12, 2022.

22 P. Erdös and R. L. Graham. On packing squares with equal squares. *Journal of Combinatorial Theory, Series A*, 19(1):119–123, 1975.

23 G. Fejes Tóth. Recent progress on packing and covering. *Contemporary Mathematics*, 223:145–162, 1999.

24 S. P. Fekete, A. Haas, P. Keldenich, M. Perk, and A. Schmidt. Computing area-optimal simple polygonalization. *Journal of Experimental Algorithmics*, 27:2.6:1–2.6:23, 2020.

25 S. P. Fekete, P. Keldenich, D. Krupke, and J. S. B. Mitchell. Area-optimal simple polygonalizations: The CG Challenge 2019. *Journal of Experimental Algorithms*, 27:1–12, 2022.

26 S. P. Fekete, P. Keldenich, D. Krupke, and J. S. B. Mitchell. Computing coordinated motion plans for robot swarms: The CG:SHOP Challenge 2021. *Journal of Experimental Algorithms*, 27:3.1:1–3.1:12, 2022.

27 S. P. Fekete, P. Keldenich, D. Krupke, and S. Schirra. Minimum partition into plane subgraphs: The CG:SHOP challenge 2022. *Journal of Experimental Algorithms*, 28:1.9:1–1.9:13, 2022.

28 S. P. Fekete, P. Keldenich, D. Krupke, and S. Schirra. Minimum coverage by convex polygons: The CG:SHOP Challenge 2023, 2023.

29 S. P. Fekete and J. Schepers. A new exact algorithm for general orthogonal d-dimensional knapsack problems. In *European Symposium on Algorithms (ESA)*, pages 144–156, 1997.

30 S. P. Fekete and J. Schepers. On higher-dimensional packing I: Modeling. Technical Report 97–288, Universität zu Köln, 1997.

31 S. P. Fekete and J. Schepers. On higher-dimensional packing II: Bounds. Technical Report 97–289, Universität zu Köln, 1997.

32 S. P. Fekete and J. Schepers. On higher-dimensional packing III: Exact algorithms. Technical Report 97–290, Universität zu Köln, 1997.

33 S. P. Fekete and J. Schepers. A general framework for bounds for higher-dimensional orthogonal packing problems. *Mathematical Methods of Operations Research*, 60:311–329, 2004.

34 S. P. Fekete, J. Schepers, and J. van der Veen. An exact algorithm for higher-dimensional orthogonal packing. *Operations Research*, 55(3):569–587, 2007.

35 F. Fontan, P. Lafourcade, L. Libralesso, and B. Momège. Local search with weighting schemes for the CG: SHOP 2022 competition. In *Symposium on Computational Geometry (SoCG)*, volume 224 of *LIPIcs*, pages 73:1–73:6, 2022.

36 T. Gensane and P. Ryckelynck. Improved dense packings of congruent squares in a square. *Discrete & Computational Geometry*, 34(1):97–109, 2005.

37 N. Goren, E. Fogel, and D. Halperin. Area-optimal polygonization using simulated annealing. *Journal of Experimental Algorithmics*, 27:2.3:1–2.3:17, 2022.

38 T. Hales, M. Adams, G. Bauer, T. D. Dang, J. Harrison, L. T. Hoang, C. Kaliszyk, V. Magron, S. McLaughlin, T. T. Nguyen, Q. T. Nguyen, T. Nipkow, S. Obua, J. Pleso, J. Rute, A. Solovyev, T. H. A. Ta, N. T. Tran, T. D. Trieu, J. Urban, K. Vu, and R. Zumkeller. A formal proof of the Kepler conjecture. *Forum of Mathematics, Pi*, 5:e2, 2017.

39 T. C. Hales. A proof of the Kepler conjecture. *Annals of Mathematics*, 162(3):1065–1185, 2005.

40 M. Held. Priority-driven nesting of irregular polygonal shapes within a convex polygonal container based on a hierarchical integer grid. In *Symposium on Computational Geometry (SoCG)*, volume 293 of *LIPIcs*, pages 85:1–85:6, 2024.

41 J. Kepler. Strena seu de nive sexangula, 1611.

42 D. M. Krupke. *Algorithm Engineering for Hard Problems in Computational Geometry*. PhD thesis, May 2022.

43 A. A. Leao, F. M. Toledo, J. F. Oliveira, M. A. Carravilla, and R. Alvarez-Valdés. Irregular packing problems: A review of mathematical models. *European Journal of Operational Research*, 282(3):803–822, 2020.

44 J. Lepagnot, L. Moalic, and D. Schmitt. Optimal area polygonization by triangulation and ray-tracing. *Journal of Experimental Algorithmics*, 27:1–23, 2022.

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

46 P. Liu, J. Spalding-Jamieson, B. Zhang, and D. W. Zheng. Coordinated motion planning through randomized k-opt. In *Symposium on Computational Geometry (SoCG)*, volume 189 of *LIPIcs*, pages 64:1–64:8, 2021.

47 A. Lodi, S. Martello, and M. Monaci. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252, 2002.

48 C. Luo, Z. Su, and Z. Lü. A general heuristic approach for maximum polygon packing. In *Symposium on Computational Geometry (SoCG)*, volume 293 of *LIPIcs*, pages 84:1–84:9, 2024.

49 A. Merino and A. Wiese. On the two-dimensional knapsack problem for convex polygons. *ACM Transactions on Algorithms*, 2020.

50 V. Milenkovic. Multiple translational containment part ii: Exact algorithms. *Algorithmica*, 19(1):183–218, 1997.

51 V. J. Milenkovic. Translational polygon containment and minimal enclosure using linear programming based restriction. In *Symposium on Theory of Computing (STOC)*, pages 109–118, 1996.

52 V. J. Milenkovic. Rotational polygon containment and minimum enclosure. In *Symposium on Computational Geometry (SoCG)*, pages 1–8, 1998.

53 V. J. Milenkovic. Rotational polygon containment and minimum enclosure using only robust 2d constructions. *Computational Geometry*, 13(1):3–19, 1999.

54 V. J. Milenkovic and K. Daniels. Translational polygon containment and minimal enclosure using mathematical programming. *International Transactions in Operational Research*, 6(5):525–554, 1999.

55 L. Moalic, D. Schmitt, J. Lepagnot, and J. Kritter. Computing low-cost convex partitions for planar point sets based on a memetic approach. In *Symposium on Computational Geometry (SoCG)*, volume 164 of *LIPIcs*, pages 84:1–84:9, 2020.

56 J. W. Moon and L. Moser. Some packing and covering theorems. In *Colloquium Mathematicae*, volume 17, pages 103–110, 1967.

57 N. Ramos, R. C. de Jesus, P. de Rezende, C. de Souza, and F. L. Usberti. Heuristics for area optimal polygonizations. *Journal of Experimental Algorithmics*, 27:2.1:1–2.1:25, 2022.

58 A. Schidler. Sat-based local search for plane subgraph partitions. In *Symposium on Computational Geometry (SoCG)*, volume 224 of *LIPIcs*, pages 74:1–74:8, 2022.

59 A. Schidler and S. Szeider. Sat-boosted tabu search for coloring massive graphs. *Journal of Experimental Algorithms*, 28:1.5:1–1.5:19, 2023.

60 J. Spalding-Jamieson, B. Zhang, and D. W. Zheng. Conflict-based local search for minimum partition into plane subgraphs. In *Symposium on Computational Geometry (SoCG)*, volume 224 of *LIPIcs*, pages 72:1–72:6, 2022.

61 P. E. Sweeney and E. R. Paternoster. Cutting and packing problems: a categorized, application-orientated research bibliography. *Journal of the Operational Research Society*, 43(7):691–706, 1992.

62 The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023.

63 G. F. Tóth. Packing and covering. In *Handbook of Discrete and Computational Geometry, Third Edition*, pages 27–66. Chapman and Hall/CRC, 2017.

64 H. Yang and A. Vigneron. A simulated annealing approach to coordinated motion planning. In *Symposium on Computational Geometry (SoCG)*, volume 189 of *LIPIcs*, pages 65:1–65:9, 2021.

65 H. Yang and A. Vigneron. Coordinated path planning through local search and simulated annealing. *Journal of Experimental Algorithms*, 27:3.3:1–3.3:14, 2022.

66 D. W. Zheng, J. Spalding-Jamieson, and B. Zhang. Computing low-cost convex partitions for planar point sets with randomized local search and constraint programming. In *Symposium on Computational Geometry (SoCG)*, volume 164 of *LIPIcs*, pages 83:1–83:7, 2020.