Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Systems
DOI: 10.54254/2755-2721/2026.GU33489

Hybrid Simulated Annealing with Machine Learning for Efficient Geometric Coverage Optimization

Yikai Zhao

Shanghai Pinghe School, Shanghai, China
zhyk201501@gmail.com

Abstract. Circle covering problems attempt to cover a given region with a fixed number of circles, and we want to make the circles have minimum radius. This paper investigates a version of the circle covering problem that is more practical, which is called geometric coverage optimization. The goal is to maximize the coverage percentage of a bounded rectangular region with a number of unit circles. The geometric coverage optimization is often applied in sensor placement, wireless network coverage, and resource allocation. This paper uses the simulated annealing algorithm to solve the problem because of its ability to escape local optima and explore a large solution space. However, if moves of circle positions are purely random, it will likely lead to high computational cost because there will be a lot of moves that fail to improve the coverage. Therefore, a lightweight machine learning model LightGBM is implemented in the search process. The model is trained to predict whether a proposed move is likely to improve coverage percentage.

Keywords: circle covering, simulated annealing, LightGBM, optimization

1. Introduction

In computational geometry, circle covering problem is a classic in the area of optimization. This kind of problem aims to determine the minimum radii of a fixed number of circles so that a specified region can be completely covered. Such problems are known to be NP-hard, which is computationally challenging [1]. However, this paper aims to tackle the circle coverage optimization problem which is a variant of circle covering. Its objective is no longer minimizing the radii of circles to fully cover a given region, but shifts to maximizing the coverage percentage of a region by a fixed number of fixed size circles. It is less challenging in computation.

The problem that this paper studies is defined as: Given a fixed rectangular region and a fixed number of unit circles with random initial positions, how to reach a high coverage percentage of the rectangle (97%) in a short time period (less than 5 minutes).

In this paper's problem setting, overlaps between circles are allowed. But to note that, excessive overlap reduces effective coverage, so it is implicitly penalized in solving the problem. Since this optimization problem is continuous, highly non-convex, and characterized by a large number of local optima, [2] simulated annealing (SA) algorithm is a great choice. It is a mechanism for escaping local minima through probabilistic acceptance of non-improving moves [3,4]. Simulated

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.GU33489

annealing has been successfully applied to a wide range of continuous geometric optimization problems, such as facility location, packing, and coverage variants [5].

Although simulated annealing is flexible, it is often computationally expensive [2]. It spends a significant portion of its runtime on evaluating candidate neighbor moves that are ultimately rejected. Especially in this kind of coverage optimization problems, this cost is further amplified because each evaluation requires coverage calculation across a large number of densely sampled points [6]. As the coverage percentage reaches a high level (for example, above 85%), the acceptance rate of improving moves drops sharply, so it leads to substantial wasted computation.

Recent advances in machine learning bring a direction for addressing the inefficiency. Machine learning models can be used to guide specific components of the search process, such as move selection or acceptance decisions [7]. In this paper, the simulated annealing is assisted by a lightweight model which is trained to predict whether a locally generated move is likely to improve coverage based on several geometric features. During optimization, the model reduces the number of costly coverage evaluations while preserving the stochastic exploration behavior of simulated annealing.

The rest of the paper is organized as follows. In Section 2 relevant literature on circle covering, continuous packing, and ML-enhanced metaheuristics is reviewed. Section 3 formalizes and defines the problem. Section 4 describes the baseline simulated annealing algorithm and the improved ML-guided method. Section 5 brings the experimental results, and section 6 the analysis and evaluations. Finally, section 7 concludes the paper.

2. Literary review

2.1. Circle covering and coverage maximization

The circle covering problem has been studied a lot in computational geometry. The traditional focus is on minimizing the number of circles required to fully cover a given region. Early work and open problems are documented in classical references such as Unsolved Problems in Geometry by Croft et al. [8] and Research Problems in Discrete Geometry by Brass et al. [9] Variants of covering problems also appear in facility location and spatial optimization literature, where geometric constraints play a central role.

Coll et al. [5] formulated placing k equal circles to cover as much of a polygonal domain as possible, and solved it by using parallel simulated annealing. They noted the problem's relation to siren/antenna placement, where each disk represents a coverage radius. In computational geometry, exact methods exist for special cases. For example, theoretical bounds for covering a rectangle with 6 or 7 circles have been derived, but these are largely of mathematical interest instead of practical algorithms. In wireless sensor networks, maximizing coverage with a fixed set of sensors (each with circular range) is also a standard NP-hard optimization task. Heuristic solutions often use greedy or sequential placement, but suffer from gaps or wasted overlap. So, exact methods for full coverage are infeasible for large instances, which motivates heuristic search.

2.2. Metaheuristics for continuous geometric optimization

Metaheuristic algorithms like simulated annealing, genetic algorithms, and particle swarm optimization have been widely implemented in continuous geometric optimization problems. Simulated annealing is particularly attractive due to its characteristic of simplicity and generality. The work by Kirkpatrick, Gelatt, and Vecchi [3] introduced simulated annealing as a general

DOI: 10.54254/2755-2721/2026.GU33489

optimization framework inspired by thermodynamic processes, which is foundational. Subsequent studies by Černý [10] and by Aarts and Korst [11] further analyzed its convergence properties and practical performance.

Huang et al. [12] addressed packing unequal circles via greedy and simulation approaches, which highlighted SA's role in related domains. Similarly, SA has been used for packing circles into fixed containers. The use of SA in this paper is inspired by the success of such works in coverage problems. As neighbor moves and cooling schedules can affect convergence speed, they often require careful tuning in metaheuristics. So, stochastic methods that can adjust circle positions continuously are more suitable than those which adjust combinatorially. The paper's approach leverages SA's ability to explore real-valued configurations of circle centers.

2.3. Machine learning guided metaheuristics optimization

Recently, machine learning has been used to enhance classical optimization algorithms. Learning-based methods are often applied to guide heuristic decisions. Several usages include predicting promising regions of the search space, selecting moves, and adapting algorithm parameters. Surveys by Bengio et al. [13] and by Mazyavkina, Nina, et al [14] highlight the growing trend and interest in combining machine learning with combinatorial and continuous optimization.

Also, neural SA and related methods use reinforcement learning to learn proposal distributions. An example is that they show improved performance over vanilla SA on benchmark problems. Szénási and Légrádi [15] systematically reviewed 48 studies of ML-augmented local search, and identified common patterns in neural-network-based move predictors and surrogate models. Many of these hybrid methods use deep learning or online training to bias the search. Kharitonov et al. [16] survey combined ML with metaheuristics and found that most work focused on ML for tuning or initialization, with relatively few studies using ML to directly prune the search space.

3. Problem definition

3.1. Initialization

Let the rectangular region be defined as:

$$R = [0, w] \times [0, h] \tag{1}$$

where w=10 and h=5.

Let n=20 identical circles of radius r=1 be given. The configuration of circles is defined by their centers:

$$C = \{(x_i, y_i)\}_{i=1}^{n} \tag{2}$$

subject to:

$$0 \leq x_i \leq w, 0 \leq y_i \leq h$$

DOI: 10.54254/2755-2721/2026.GU33489

3.2. Coverage approximation

Coverage is approximated via uniform grid sampling. A grid of N=num_points2 points is generated over the rectangle, where num_points=150. A grid point p is considered covered if:

$$\exists i \quad s.t. \quad \|p - c_i\|^2 \leq r^2 \tag{3}$$

Coverage percentage is defined as:

$$Coverage\,(C) = 1 - \frac{UncoveredPoints}{N} \tag{4}$$

3.3. Objective

The goal of the experiment in this paper is to make coverage percentage C reach 0.97 in a relatively short time period.

4. Algorithms

Note: The algorithm code can be found through the Github link in the appendix.

4.1. Simulated annealing algorithm

Simulated annealing is used as the baseline metaheuristic. At each iteration modification is made to a randomly selected circle center. The move of the x and y coordinates of the chosen circle is within the step size of 0.5. The change in cost function is then computed, and if the cost function decreases, the move is accepted deterministically; otherwise, it is accepted with a probability of exp(−ΔF/T) where ΔF is the increase in cost and T is the current temperature. The temperature is gradually decreased according to a cooling schedule T ← γT with γ=0.995 [17,18].

4.2. Problem initialization

Code: see appendix.

4.3. Uncovered point count

One key efficiency bottleneck is the cost of checking each grid point whether it falls inside any of the circles. The traditional method of double looping over points and circles is slow for large grid sizes. To address this, NumPy vectorization and broadcasting is implemented to compute distances between all grid points and all circle centers in a single array operation. Let N denote the number of grid points and C the number of circles, and grid points are arranged in an (N,2) array and circle centers in a (C,2) array. Broadcasting computes a (N,C,2) difference tensor which is squared and summed along the coordinate axis to produce a (N,C) matrix of squared distances. A logical OR-reduction along the circle axis then identifies which grid points are covered. The uncovered point count is simply the number of grid points not covered by any circle. This approach reduces Python-level looping overhead and yields a substantial speed-up.

DOI: 10.54254/2755-2721/2026.GU33489

4.4. Overlap penalty function

To discourage excessive local overlap between circles, the cost function includes a penalty for overlap between circles. For two circles with radii r_i and r_j and centers separated by distance d, a simple overlap measure is max(0, r_i + r_j − d). Squaring is chosen to more strongly penalize large overlaps. The total cost thus combines the uncovered point count with a weighted overlap penalty:

$$OverlapPenalty = \sum_{i<j} \max\left(0, r_i + r_j - d_{ij}\right)^2 \tag{5}$$

$$f = \alpha \bullet UncoveredPoints + \beta \bullet OverlapPenalty \tag{6}$$

with α=1 and β=5.

4.5. The features and procedures of the LightGBM model

In order to predict the expected change in the cost function after a move, a LightGBM regression model is trained. Training data is collected from 5000 random sample moves. Each data includes the original circle center coordinates (cx, cy), the proposed displacement (dx, dy), the distances from the circle center to the four rectangle boundaries (dist_left, dist_right, dist_bottom, dist_top). It also includes local density, which is defined as the number of other circles that overlap with the current circle. These features help predict whether a proposed move will produce a positive change in cost. The label construction for regression is straightforward. For each sampled move, the label is the signed change in total cost ΔF = F(new_configuration) − F(old_configuration). A negative ΔF indicates that the move is increasing the coverage percentage.

4.6. Integrating LightGBM with simulated annealing

At each iteration, 5 candidate modifications are generated. The feature vectors for all candidates are computed and fed in batch to the LightGBM model, which outputs predicted ΔF values for each candidate. The candidates are then ranked according to predicted improvement and the top is selected for actual move. The acceptance decision still relies on the true cost computed, and the rest is the baseline simulated annealing algorithm explained in 4.1. In this way, the optimization search goes toward moves the model believes are promising while retaining the stochastic exploration capability of simulated annealing that allows occasional uphill moves for escaping local optima.

5. Results

The algorithm was repeated 10 times with 10 different random initialized circle configurations, and the results are as follows:

Table 1: Trial results

Trial number	Initial coverage percentage	Final coverage percentage	Time taken (s)
1	0.735	0.971	109.5
2	0.599	0.971	78.9
3	0.728	0.972	104.2
4	0.662	0.971	100.9

DOI: 10.54254/2755-2721/2026.GU33489

Table 1. (continued)

5	0.642	0.971	116.8
6	0.709	0.971	470.1
7	0.690	0.971	230.8
8	0.609	0.970	149.2
9	0.659	0.970	67.7
10	0.650	0.980	387.6

The figures below show the initial and resulted configurations of all experiment trials.

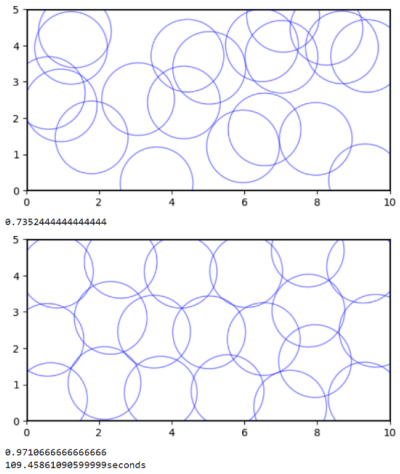

Figure 1. a: Result for trial 1

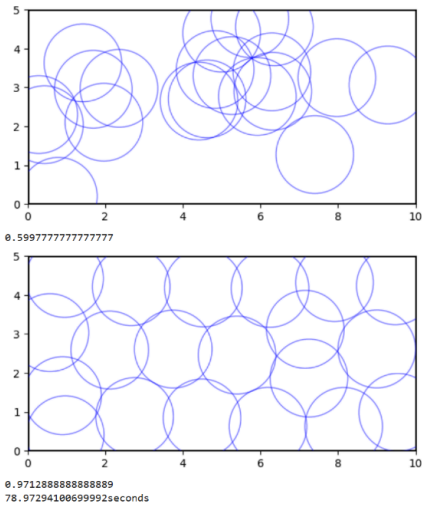

Figure 1. b: Result for trial 2

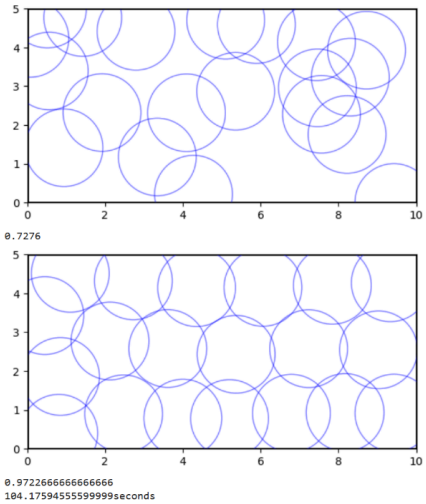

Figure 1. c: Result for trial 3

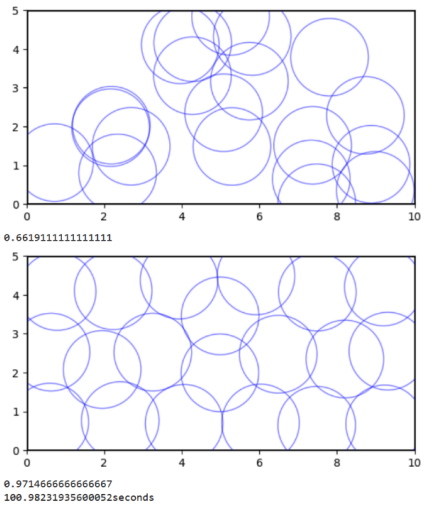

Figure 1. d: Result for trial 4

DOI: 10.54254/2755-2721/2026.GU33489

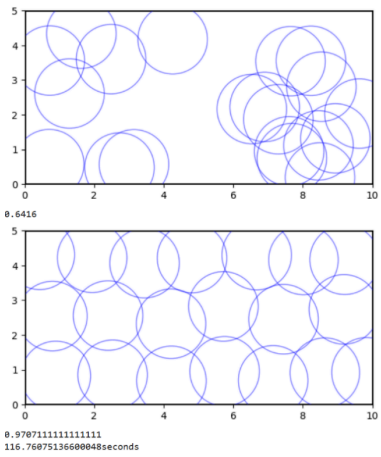

Figure 1. e: Result for trial 5

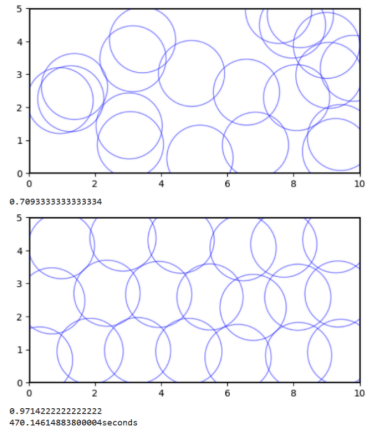

Figure 1. f: Result for trial 6

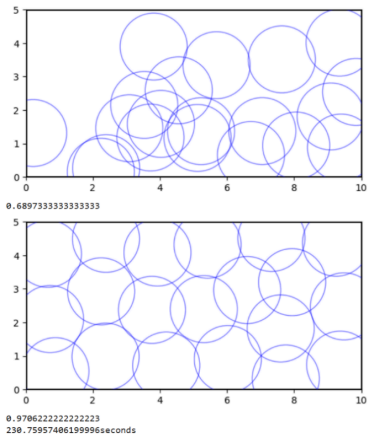

Figure 1. g: Result for trial 7

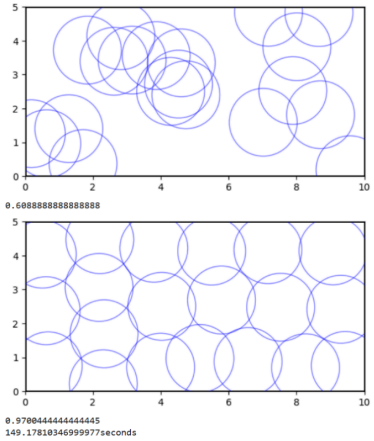

Figure 1. h: Result for trial 8

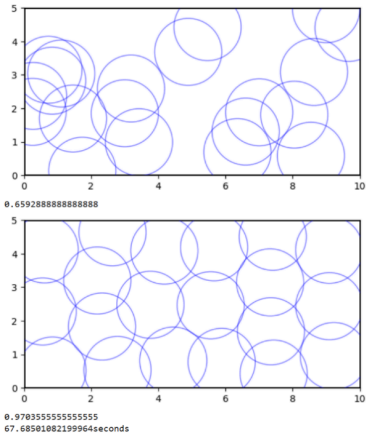

Figure 1. i: Result for trial 9

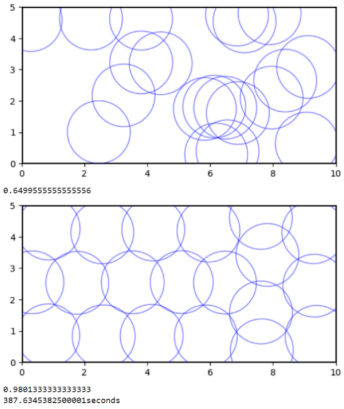

Figure 1. j: Result for trial 10

DOI: 10.54254/2755-2721/2026.GU33489

6. Analysis and evaluation

The hybrid simulated annealing algorithm with LightGBM can consistently achieve high coverage percentage across different initial configurations in short time periods. This is demonstrated by the experiment results. In all 10 trials, the final coverage percentage reached at least 0.970, with two runs achieving 0.972 and 0.980. And, the time taken is generally within the range of 4 minutes. This indicates that the algorithm is robust with respect to initialization, and it is able to reliably converge to high-quality solutions in a highly non-convex search space.

A key observation from the results is the variation in runtime. In Table 1, while some runs converged in approximately 70–120 seconds, others took significantly more time (e.g., 470.1 seconds). This difference can be explained by the stochastic nature of simulated annealing. The trajectory of the search depends on random initial configurations and probabilistic acceptance decisions. The variability is in fact consistent with prior studies, which highlight that convergence speed can fluctuate due to differences in exploration paths and cooling dynamics [19].

The LightGBM model plays an important role in improving efficiency. The algorithm reduces the number of expensive coverage evaluations by selecting the most promising candidate move among multiple choices. This is important, especially in later stages of optimization where naïve random exploration leads to a lot of wasted computation. Reaching a high coverage within a relatively short time period as observed suggests that the approach successfully mitigates this inefficiency.

This result aligns with findings in literature about hybrid optimization. A literature review mentioned the strength: machine learning guides search decisions using learned patterns, while metaheuristics maintain global exploration [20]. In this work, the LightGBM model filters low-quality moves before full evaluation, which effectively reduces computational overhead without significantly compromising solution quality.

However, several limitations should be noted. First, the model is trained offline using randomly generated sample moves. As a result, it may not fully capture the distribution of states encountered during later stages of optimization, when the coverage percentage becomes relatively high. This matters because configurations later on become more structured. To solve this, online or adaptive training could be used to improve prediction accuracy and performance. Second, although the model reduces unnecessary evaluations, it introduces additional overhead because of feature computation and prediction. Although this overhead is relatively small compared to coverage evaluation, it may become significant if the number of candidate moves increases.

Another limitation is in the approximation of coverage. While a grid of 150×150 points that this paper uses provides a reasonable trade-off between accuracy and computational cost, it introduces discretization error. Increasing grid resolution would improve accuracy but also significantly increase computational costs.

Overall, the results indicate that the hybrid approach balances exploration and efficiency. Simulated annealing ensures global search capability and robustness to local optima, and the machine learning component accelerates convergence. This combination is suitable for continuous geometric optimization problems with expensive objective evaluations.

7. Conclusion

This paper investigated the geometric coverage optimization problem, in which we change the positions of a fixed number of identical circles so that the coverage of a rectangular can reach a high number. The problem has a continuous and highly non-convex nature, so simulated annealing is chosen as the baseline optimization method. Then, a LightGBM model is also added to reduce

DOI: 10.54254/2755-2721/2026.GU33489

inefficiency of random neighbor selection in simulated annealing. The model predicts the expected improvement of candidate moves based on geometric features. The process prioritizes promising modifications while retaining the probabilistic acceptance at the same time.

Experimental results demonstrate that the method consistently achieves high coverage levels (≥97%) across multiple trials. At the same time, the total runtime of each trial is ideal. As you can see, this approach improves overall efficiency compared to standard simulated annealing because it reduces the number of unwanted move calculations, especially in later stages of the optimization.

The study also shows the potential of combining machine learning with metaheuristic in optimization. As recent literature suggested, hybrid methods can be effective in overcoming the limitations of individual approaches. It is done by leveraging learned knowledge to guide search processes [20]. This makes them really suitable for complex optimization problems with expensive cost functions.

Future work may explore these several directions for improvement. First, online learning could allow the model to adapt dynamically during optimization. Second, more advanced models such as neural networks or reinforcement learning models could further enhance decision-making. Furthermore, alternative coverage calculation methods or adaptive sampling strategies could reduce approximation error.

In conclusion, the framework and algorithms proposed in this paper demonstrates a practical and effective approach for solving continuous geometric coverage optimization problems, which provides a promising direction for further research and improvements in this area of study.

References

[1] Hifi, Mhand, and Rym M′ Hallah. "A literature review on circle and sphere packing problems: Models and methodologies." Advances in Operations Research 2009.1 (2009): 150624.
[2] Suman, Balram, and Prabhat Kumar. "A survey of simulated annealing as a tool for single and multiobjective optimization." Journal of the operational research society 57.10 (2006): 1143-1160.
[3] Kirkpatrick, Scott, C. Daniel Gelatt Jr, and Mario P. Vecchi. "Optimization by simulated annealing." science 220.4598 (1983): 671-680.
[4] Johnson, David S. "Optimization by simulated annealing: an experimental evaluations (part 1 & 2)." Tech. Report AT & Bell Laboratories (1987).
[5] Coll, Narcis, Marta Fort, and Moises Saus. "Coverage area maximization with parallel simulated annealing." Expert Systems with Applications 202 (2022): 117185.
[6] Jia, Fengman, and D. Lichti. "A comparison of simulated annealing, genetic algorithm and particle swarm optimization in optimal first-order design of indoor TLS networks." ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 4 (2017): 75-82.
[7] Kosanoglu, Fuat, Mahir Atmis, and Hasan Hüseyin Turan. "A deep reinforcement learning assisted simulated annealing algorithm for a maintenance planning problem." Annals of Operations Research 339.1 (2024): 79-110.
[8] Croft, Hallard T., Kenneth Falconer, and Richard K. Guy. Unsolved problems in geometry: unsolved problems in intuitive mathematics. Vol. 2. Springer Science & Business Media, 2012.
[9] Brass, Peter, William OJ Moser, and János Pach. Research problems in discrete geometry. New York, NY: Springer New York, 2005.
[10] Černý, Vladimír. "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm." Journal of optimization theory and applications 45.1 (1985): 41-51.
[11] Aarts, Emile HL, Jan HM Korst, and Peter JM van Laarhoven. "A quantitative analysis of the simulated annealing algorithm: A case study for the traveling salesman problem." Journal of Statistical Physics 50.1 (1988): 187-206.
[12] Huang, Wen Qi, et al. "Greedy algorithms for packing unequal circles into a rectangular container." Journal of the Operational Research Society 56.5 (2005): 539-548.
[13] Bengio, Yoshua, Andrea Lodi, and Antoine Prouvost. "Machine learning for combinatorial optimization: a methodological tour d'horizon." European Journal of Operational Research 290.2 (2021): 405-421.

DOI: 10.54254/2755-2721/2026.GU33489

[14] Mazyavkina, Nina, et al. "Reinforcement learning for combinatorial optimization: A survey." Computers & Operations Research 134 (2021): 105400.
[15] Szénási, Sándor, Gábor Légrádi, and Balàzs Vígh. "Machine learning-assisted approach for optimizing step size of hill climbing algorithm." 2024 IEEE 18th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2024.
[16] Kharitonov, André, et al. "Literature survey on combining machine learning and metaheuristics for decision-making." Procedia Computer Science 253 (2025): 199-208.
[17] Dalfard, Vahid Majazi. "Adjustment of the primitive parameters of the simulated annealing heuristic." (2011).
[18] Zhang, D., and Xin Li. "A personified annealing algorithm for circles packing problem." Acta Automatica Sinica 31.4 (2005): 590.
[19] Laarhoven, Peter JM, and Emile HL Aarts. Simulated annealing: theory and applications. Dordrecht: D. Reidel, 1987.
[20] Azevedo, Beatriz Flamia, Ana Maria AC Rocha, and Ana I. Pereira. "Hybrid approaches to optimization and machine learning methods: a systematic literature review." Machine Learning 113.7 (2024): 4055-4097.

Appendix

The full code and the code corresponding to each sub-section in the "4. Algorithms" section can be found in my personal Github repository called "jerrymcnba/coverage_optimization_SA-ML". The link to the repository is as follows:

https://github.com/jerrymcnba/coverage_optimization_SA-ML