DeepMCLP: Solving the MCLP with Deep Reinforcement Learning for Urban Spatial Computing*

Shaohua Wang[1,2], Haojian Liang[3], Yang Zhong[4], Xueyan Zhang[5], and Cheng Su[1,2]

[1] International Research Center of Big Data for Sustainable Development Goals, Beijing 100094, China
[2] State Key Laboratory of Remote Sensing Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
wangshaohua@aircas.ac.cn
[3] School of Artificial Intelligence, Jilin University, Changchun 130012, China
[4] School of Information Systems and Technology, Claremont Graduate University, Claremont, CA 91711, USA
[5] Viterbi School of Engineering, University of Southern California, Los Angeles, CA 90089, USA

Abstract. Maximal Covering Location Problem (MCLP) is a classical spatial optimization problem that plays a significant role in urban spatial computing. Due to its NP-hard, finding an exact solution for this problem is computationally challenging. This study proposes a deep reinforcement learning-based approach called DeepMCLP to address the MCLP problem. We model MCLP as a Markov Decision Process. The encoder with attention mechanisms learns the interaction between demand points and facility points and the decoder outputs a probability distribution over candidate facility points, and a greedy policy is employed to select facility points, resulting in a feasible solution. We utilize the trained DeepMCLP model to solve both artificially synthesized data and real-world scenarios. Experimental results demonstrate that our algorithm effectively solves the MCLP problem, achieving faster solving times compared to mature solvers and smaller optimality gaps compared to the genetic algorithm. Our algorithm offers a novel perspective on solving spatial optimization problems, and future research can explore its application to other spatial optimization problems, providing scientific and effective guidance for urban planning and urban spatial analysis.

Keywords: Urban Spatial Computing · Spatial Optimization Problem · Maximum Covering Location Problem · Attention Mechanism · Deep Reinforcement Learning.

DOI: https://doi.org/10.25436/E2KK5V.

* Supported by The Chaoyang District Collaborative Innovation Project (E2DZ050100) and the Hundred Talents Program Youth Project (Category B) of the Chinese Academy of Sciences (E2Z10501)

1 Introduction

Urban spatial computing is a methodological approach that investigates the characteristics, patterns, and challenges of urban spaces. It involves the quantitative analysis and evaluation of spatial attributes in cities using techniques such as geographic information systems (GIS), statistics, and spatial data analysis. Urban spatial optimization is a vital component of urban spatial computing, where varying constraint conditions are applied to achieve optimized layouts and minimize costs or maximize objective functions based on real-world scenarios. Spatial optimization provides decision-makers with scientifically sound solutions, playing a crucial and positive role in urban planning, transportation optimization, and sustainable urban development [9, 11, 12, 18].

The discrete facility location problem is a well-known NP-hard problem in operations research and is one of the most representative spatial optimization problems. Among them, the MCLP stands out as a fundamental problem. It was originally proposed by Church [4] and has found extensive applications in logistics management, urban planning, and other related domains [5, 19–21]. In the context of the maximum coverage problem, given a set of demand points and candidate facility points, the objective is to select a certain number of facilities from the candidates to maximize the coverage of the demand points. Typically, each facility has a maximum service distance that defines the reachability between demand points and facility points.

Research on the MCLP has yielded numerous approaches for solving it, including exact algorithms [7], approximation algorithms [6], and heuristic algorithms [1, 8, 10]. Exact algorithms provide optimal solutions for small-scale instances but are impractical for large-scale problems. Approximation algorithms offer suboptimal solutions, with a theoretical bound on the gap between the suboptimal and optimal solutions known as the approximation ratio α. Heuristic methods provide fast solutions for MCLP but do not guarantee optimality. MCLPs are commonly formulated as mixed-integer linear programming (MILP), making solver-based approaches viable. Prominent solvers, such as Gurobi, Cplex, OR-tools, SCIP, COPT, are available, with SCIP being an open-source option. These solvers employ specialized algorithms and heuristics to rapidly and accurately solve MCLP within certain problem sizes.

Despite the existence of numerous algorithms for solving the MCLP, the problem's NP-hard nature precludes finding exact solutions. In recent years, deep learning models have demonstrated their ability to extract meaningful features [13–15, 17]. Consequently, we propose a novel algorithm for efficiently solving the MCLP problem using deep reinforcement learning. By leveraging attention mechanisms to capture the interactions between demand points and facility points, our algorithm trains a deep learning model that directly solves the MCLP problem. Our approach outperforms genetic algorithms in terms of solution accuracy while maintaining computational efficiency. To evaluate the algorithm's reliability, we conduct experiments on both synthetic and real-world datasets. The experimental results validate the effectiveness of our algorithm in solving MCLP and highlight its significant contributions to urban spatial optimization, facility layout, and sustainable urban development. The main contributions of this paper include:

– Introduction of a novel algorithm based on deep reinforcement learning and attention mechanisms for MCLP solving.
– Evaluation of the algorithm's performance on both synthetic and real-world datasets.
– Comparative analysis demonstrating the superiority of our proposed algorithm over genetic algorithms.
– Provision of valuable insights and guidance for urban spatial optimization, facility layout, and sustainable urban development.

This research comprises six major sections. The second section introduces the preliminary works of the study area and problem definition. The third section outlines the research methodology and algorithm design. In the fourth section, experiments are conducted using both artificially synthesized data and real-world scenarios. The fifth section discusses the results and findings obtained from the experiments. Finally, the last section presents the conclusion of the study and outlines potential avenues for future research.

2 Preliminaries

2.1 Study Area

Seattle, situated in the state of Washington, is a city that holds significant importance as the largest urban center in the state and a vital economic, cultural, and technological hub in the Pacific Northwest region. Renowned for its dynamic atmosphere and spirit of innovation, Seattle is home to a flourishing tech industry, a wealth of cultural activities, and awe-inspiring natural landscapes, attracting individuals from diverse backgrounds worldwide. Given its high population density, diverse demographics, and bustling transportation networks, Seattle serves as an optimal research area.

Figure 1 depicts the visual representation of population data and the spatial distribution of candidate facility points in Seattle. The primary objective is to select a predetermined number of billboards from the candidate billboards in order to maximize advertisement exposure or optimize the profitability for advertisers. This problem can be described as a maximum covering location problem.

2.2 Problem Definition

The MCLP involves two types of nodes: demand points and facility points. The goal is to select a specific number of facilities from the candidate facility points to maximize the coverage of demand points. This problem is commonly represented using a bipartite graph, denoted as $G = (U, F, E)$. Here, $U = \{1, 2, ..., N\}$ represents the set of all demand points, $F = \{1, 2, ..., M\}$ represents the set of all candidate facility points, and $E = \{(i, j) : i \in U, j \in F\}$ represents the set of all

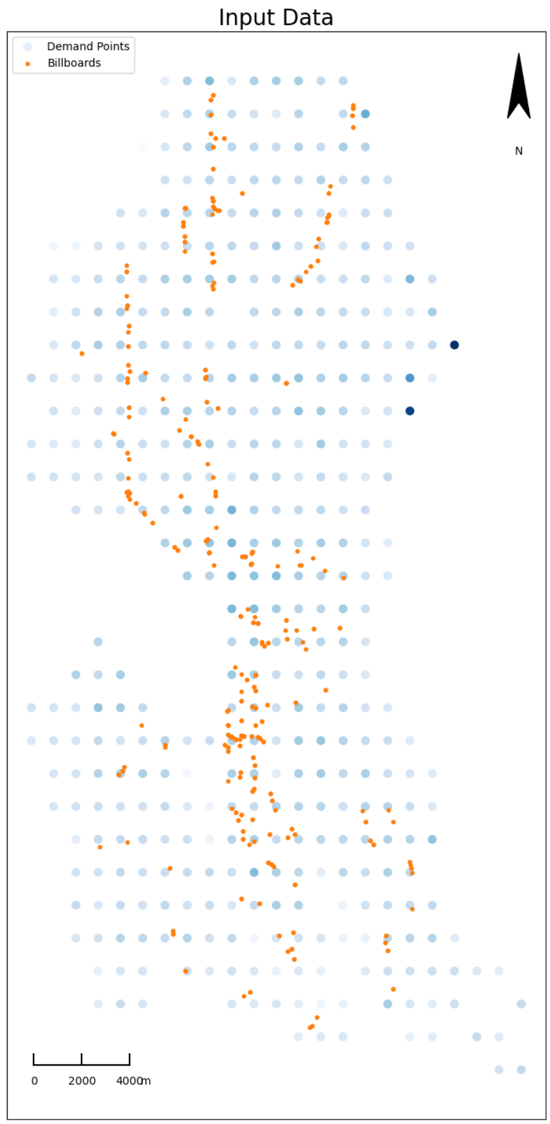


Fig. 1. The spatial units of Seattle City. The blue dots represent population points, with the intensity of color varying based on the quantity of demand (or flow) – the greater the demand, the deeper the color. The orange dots indicate candidate locations for billboards, and these candidate points' positions are determined by POI data.

edges. The demand quantity or population is denoted as d, while v_{ij} represents the distance between demand point i and facility j. Additionally, a constant value S is introduced to represent the coverage radius of the nodes. If $v_{ij} < S$, it signifies that demand point i is covered by facility j. To express the coverage status of demand points, a binary variable b is introduced.

$$b_i = \begin{cases} 1, & \text{if demand point } i \text{ is covered,} \\ 0, & \text{otherwise.} \end{cases}$$

a is also a binary variable used to indicate whether the facility is selected.

$$a_j = \begin{cases} 1, & \text{if facilities } j \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

The objective function of the MCLP can be expressed as follows:

$$max \sum_{i=1}^{N} d_i b_i$$

In general, the MCLP can be formulated as a MILP problem, which can be represented as follows [4]:

$$Maximize \qquad z = \sum_{i=1}^{N} d_i b_i \tag{1}$$

$$s.t. \qquad \sum_{j \in N_i} a_j \geq b_i, i \in U \tag{2}$$

$$\sum_{j=1}^{M} a_j = p \tag{3}$$

$$b_i \in \{0, 1\}, i \in U \tag{4}$$

$$a_j \in \{0, 1\}, j \in F. \tag{5}$$

- U: Set of demand points
- F: Set of candidate facilities
- S: Maximum service distance for candidate facilities
- v_{ij}: Distance from node i to node j
- $N_i = \{j \in F | v_{ij} \leq S\}$
- d_i: Population quantity of demand point i
- p: Number of facilities to be located

3 Methodology

3.1 Workflow

This study introduces a novel approach for solving the MCLP. Leveraging the inherent characteristics of the problem, we utilize deep learning models to uncover the intricate interactions between demand points and facility points. The models are trained using deep reinforcement learning algorithms. The workflow of our proposed solution framework is depicted in Figure 2.

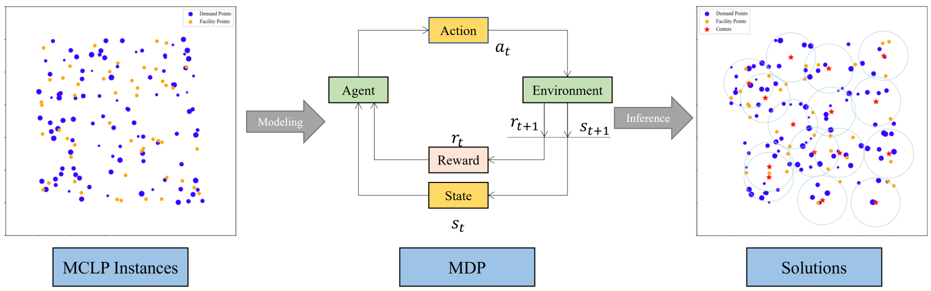


Fig. 2. The workflow of our proposed solution framework.

The fundamental challenge in the MCLP is determining the optimal strategy to select p facility points in order to maximize the coverage of demand points. To address this, we adopt a constructive approach for generating solutions. We formulate the problem as a Markov Decision Process (MDP) and leverage deep reinforcement learning algorithms to train the deep learning models. These trained models guide decision-makers in the selection of facility points at each step, ultimately leading to the generation of the final solution. The deep learning model consists of an Encoder-Decoder structure, where the encoder and decoder incorporate multi-head attention layers.

Section 3.2 provides a comprehensive explanation of how the MCLP can be modeled as an MDP process. In Section 3.3, we delve into the prominent attention mechanisms utilized in the Encoder and Decoder. Furthermore, Section 3.4 elucidates the training process of deep reinforcement learning.

3.2 Markov Decision Process

MDP is a mathematical model used to describe a stochastic process for decision problems. In an MDP, a decision problem is modeled as a process consisting of states, actions, rewards, and transition probabilities. By defining states, actions, rewards, and transition probabilities, an MDP can be formalized as a quintuple (S, A, P, R, γ), where

- S represents the current state of demand points and facility points.
- A represents the current selected facility point.
- $P(s'|s, a)$ is the transition probability of the system moving from state S to state S' when action A is taken.
- $R(s, a, s')$ is the reward obtained when transitioning from state S to state S' by taking action A.
- γ is the discount factor, which balances the importance of immediate rewards and future rewards.

3.3 Attention Mechanism

The attention mechanism [22] is one of the most widely used techniques in deep learning, which aims to emulate human attention behavior and capture deep-level features between nodes. It typically involves three essential components: query Q, key K, and value V. By computing the similarity between the query and key, attention weights are assigned to each key. These weights are then normalized to ensure their sum equals 1. Finally, the values are weighted by the attention weights and aggregated. The attention mechanism effectively handles complex relationships between demand points and facility points, capturing crucial information from input sequences and significantly enhancing the model's performance. The basic formula for calculating attention is as follows:

$$Att(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

The scaling factor, denoted as d_k, is related to the dimension of the query Q.

In our DeepMCLP model, we employ the attention mechanism to facilitate the interaction learning among nodes. The encoder encodes the complex relationships between nodes, and its output is fed into the decoder. The decoder generates a probability distribution over all candidate facility points. To select the central point, a greedy strategy is employed, choosing the node with the highest probability. This approach enables the model to effectively capture the interplay between nodes and make informed decisions regarding the selection of facility points.

3.4 Deep Reinforcement Learning

Unlike supervised learning, reinforcement learning [3] does not require labeled data during the training process. Instead, it involves the interaction between an agent and its environment, where the agent learns an optimal policy based on the rewards or feedback received from the environment. In the context of MCLP, we utilize the objective function of the problem as the feedback signal. Specifically, we aim to train our model to select p facility points in p steps that maximize the number of serviced demand points. We primarily employ the REINFORCE [16] algorithm to train the DeepMCLP model, enabling it to learn the optimal policy and achieve the highest cumulative reward. The pseudo code for the training process is shown in Algorithm 1.

4 Experiments and Results

In our study, we have presented the main methodology for solving the MCLP problem. As deep learning models are often considered as black-box models with limited interpretability, we conducted experiments using synthetic data and real-world scenarios to assess the effectiveness of our algorithm. In addition to our proposed DeepMCLP approach, we also implemented Gurobi solver [16] and a

Algorithm 1 Training the Deep Models for MCLP by Reinforcement Learning

Input: Training data S, Number of epochs E, Number of training steps P, Batch size B

Initialize the network parameters θ
for epoch = 1: E **do**
    for step = 1: P **do**
        $s_i \leftarrow$ SampleInput(S) for $i \in \{1, ..., B\}$
        $\pi_i \leftarrow p_\theta(s_i)$. The policy by sampling
        Calculate the objective function $\mathrm{L}(\pi_i)$
        Calculate the gradient, $\nabla\mathcal{L} \leftarrow \sum_{i=1}^{B}(\mathrm{L}(\pi_i))\nabla_\theta log p_\theta(\pi_i)$
        Update the parameters, $\theta \leftarrow \theta + \nabla\mathcal{L}(\theta)$
    end for
end for

genetic algorithm [2] to solve the MCLP problem. Through extensive analysis of the experimental results, we were able to gain insights and make comparisons among these different approaches.

4.1 Synthetic Data

The DeepMCLP model requires a certain amount of time for training. Once the model training is completed, it enables rapid solving of MCLP instances of various sizes. Therefore, we conducted experiments using a dataset generated on a 1x1 plane, consisting of a large number of 200 demand points and 100 facility points. Subsequently, we tested the performance of the Gurobi solver, DeepMCLP, and genetic algorithm on different MCLP instances. Table 1 presents the results, including the objective function value, the gap between the obtained solution and the optimal solution, and the solution time.

Table 1. The results of Gurobi, GA, and DeepMCLP for solving MCLP in synthetic data

Algorithm	N=50, M=20, p=8, R=0.2			N=100, M=50, p=15, R=0.15			N=200, M=100, p=25, R=0.1		
	Obj.	Gap	Time/s	Obj.	Gap	Time/s	Obj.	Gap	Time/s
Gurobi	248	0.00%	0.0715	537	0.00%	0.1880	926	0.00%	0.9670
GA	152	38.71%	0.0609	437	18.62%	0.0866	707	23.65%	0.2059
DeepMCLP	222	10.48%	0.0052	518	3.53%	0.0121	810	12.53%	0.0214

4.2 Real Scenario

To validate the practical applicability of our model, we conducted experiments in the Seattle area. We employed a grid-based approach to discretize the population data, resulting in 428 demand points. Combining this with POI data, we obtained 417 locations for commercial billboards. The objective was to select 15 and 25 nodes from the pool of 417 billboards to maximize the coverage of demand points. Table 2 presents the results obtained using three different algorithms: the Gurobi solver, the genetic algorithm and DeepMCLP. The table showcases the objective function values, the gap between the obtained solutions and the optimal solutions, and the solution times. Furthermore, we provide visualizations of the results for the three algorithms. Figure 3 depicts the visual representation of the solutions obtained by each algorithm when selecting 15 nodes, while Figure 4 showcases the solutions for selecting 25 nodes.

Table 2. The results of Gurobi, GA and DeepMCLP for solving MCLP with p=15 in real scenario

Algorithm	N=428,M=417, p=15, R=1000			N=428,M=417, p=15, R=2000			N=428,M=417, p=15, R=3000		
	Obj.	Gap	Time/s	Obj.	Gap	Time/s	Obj.	Gap	Time/s
Gurobi	1358	0.00%	3.7458	3683	0.00%	3.8135	6529	0.00%	3.6023
GA	1135	16.42%	0.1264	3463	5.97%	0.1228	5886	9.85%	0.1288
DeepMCLP	1280	5.74%	0.1023	3612	1.93%	0.101	6473	0.86%	0.1023

Table 3. The results of Gurobi, GA and DeepMCLP for solving MCLP with p=25 in real scenario

Algorithm	N=428, M=417, p=25, R=1000			N=428, M=417, p=25, R=2000			N=428, M=417, p=25, R=3000		
	Obj.	Gap	Time/s	Obj.	Gap	Time/s	Obj.	Gap	Time/s
Gurobi	2049	0.00%	3.5055	5289	0.00%	3.6193	7615	0.00%	3.7743
GA	1722	15.96%	0.1826	4838	8.53%	0.2424	7127	6.41%	0.1856
DeepMCLP	1832	10.60%	0.1816	5241	0.91%	0.1596	7554	0.80%	0.1473

5 Discussion

In this section, we further discuss the main results from three perspectives: the performance of different algorithms, synthetic data and real-world scenarios, and the limitations of DeepMCLP.

5.1 Performance of Different Algorithms

We conducted comparative experiments between DeepMCLP, Gurobi solver, and GA to solve MCLP instances with different scales, represented by (N, M, p, R)

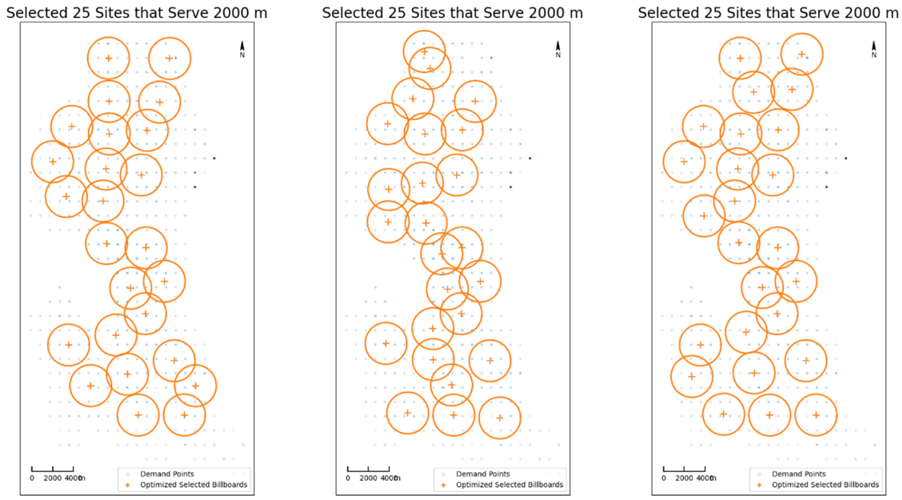
Fig. 3. (a) represents the solution obtained by the Gurobi solver for the MCLP problem with P=15 and R=2000; (b) represents the solution obtained by the GA for the MCLP problem with P=15 and R=2000;(c) represents the solution obtained by the DeepMCLP for the MCLP problem with P=15 and R=2000.

values of $(50, 20, 8, 0.2)$, $(100, 50, 15, 0.15)$, and $(200, 100, 25, 0.1)$. Here, N represents the number of demand points, M represents the number of facility points, p represents the number of selected facility points, and R denotes the coverage radius of the facilities.

For these problem instances, the Gurobi solver consistently provided fast and accurate solutions, which were considered as the optimal solutions for calculating the optimality gaps of the genetic algorithm and our proposed algorithm. Our algorithm consistently outperformed both the Gurobi solver and the genetic algorithm in terms of solution time. It demonstrated faster computation, offering significant time savings. In terms of the optimality gaps compared to the optimal solutions, DeepMCLP outperformed the genetic algorithm, achieving smaller gaps.

5.2 Synthetic Data and Real-World Scenarios

Our model was trained on synthesized data, which resulted in better performance on synthetic data. However, we also applied our model to real-world scenarios and compared it with the Gurobi solver and genetic algorithm. Although the genetic algorithm provided a quick solution, there still existed a significant gap between its solution and the optimal solution. In contrast, the Gurobi solver performed exceptionally well on these small-scale cases.



(a) (b) (c)

Fig. 4. (a) represents the solution obtained by the Gurobi solver for the MCLP problem with P=25 and R=2000; (b) represents the solution obtained by the GA for the MCLP problem with P=25 and R=2000;(c) represents the solution obtained by the DeepMCLP for the MCLP problem with P=25 and R=2000.

5.3 Limitation of DeepMCLP

Although our model demonstrates excellent performance on synthetic data, applying it to real-world problem solving still presents certain challenges. This is primarily due to the inherent differences in data distribution between actual scenarios and the training data used for the model. Consequently, for a specific real-world context, it would be advantageous to train a model using historical data specifically tailored to that particular scenario. This approach would enhance the accuracy and precision of the solutions obtained.

6 Conclusions and Future Works

The MCLP is a crucial spatial optimization problem with wide applications in selecting public facilities like parks and hospitals. It also plays a significant role in emergency facility placement, providing valuable insights for urban planning and sustainable development. However, due to its NP-hard nature, there is no algorithm capable of solving MCLP optimally. In this study, we propose a novel deep learning-based algorithm to address this challenge. Our approach leverages attention mechanisms to capture the complex interactions between demand points and facility points. By employing deep reinforcement learning, we train the model to learn the optimal policy for selecting facility points to maximize coverage. Once the model is trained, it enables efficient and fast solutions for MCLP across various problem sizes. Experimental evaluations conducted on synthetic data and real-world scenarios demonstrate the effectiveness of our algorithm. Compared to the Gurobi solver, our approach achieves faster solution times, and when compared to the genetic algorithm, it exhibits smaller gaps with the optimal solutions. For future work, there are several potential directions to explore. Firstly, incorporating additional constraints such as facility capacity and budget limitations can enhance the algorithm's applicability. Secondly, investigating scalability to handle larger-scale MCLP instances would be valuable. Lastly, conducting extensive case studies in diverse real-world applications will further validate the algorithm's robustness and performance.

References

1. Adenso-Diaz, B., Rodriguez, F.: A simple search heuristic for the mclp: Application to the location of ambulance bases in a rural region. Omega **25**(2), 181–187 (1997)
2. Anand, R., Aggarwal, D., Kumar, V.: A comparative analysis of optimization solvers. Journal of Statistics and Management Systems **20**(4), 623–635 (2017)
3. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine **34**(6), 26–38 (2017)
4. Church, R., ReVelle, C.: The maximal covering location problem. In: Papers of the regional science association. vol. 32, pp. 101–118. Springer-Verlag Berlin/Heidelberg (1974)
5. Church, R.L., Wang, S.: Solving the p-median problem on regular and lattice networks. Computers & Operations Research **123**, 105057 (2020)
6. Coco, A.A., Santos, A.C., Noronha, T.F.: Formulation and algorithms for the robust maximal covering location problem. Electronic Notes in Discrete Mathematics **64**, 145–154 (2018)
7. Cordeau, J.F., Furini, F., Ljubić, I.: Benders decomposition for very large scale partial set covering and maximal covering location problems. European Journal of Operational Research **275**(3), 882–896 (2019)
8. Davari, S., Zarandi, M.H.F., Turksen, I.B.: A greedy variable neighborhood search heuristic for the maximal covering location problem with fuzzy coverage radii. Knowledge-Based Systems **41**, 68–76 (2013)
9. Fritze, R., Graser, A., Sinnl, M.: Combining spatial information and optimization for locating emergency medical service stations: A case study for lower austria. International Journal of Medical Informatics **111**, 24–36 (2018)
10. Galvão, R.D., ReVelle, C.: A lagrangean heuristic for the maximal covering location problem. European Journal of Operational Research **88**(1), 114–123 (1996)
11. Hu, Y., Gao, S., Janowicz, K., Yu, B., Li, W., Prasad, S.: Extracting and understanding urban areas of interest using geotagged photos. Computers, Environment and Urban Systems **54**, 240–254 (2015)
12. Janowicz, K., Gao, S., McKenzie, G., Hu, Y., Bhaduri, B.: Geoai: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond (2020)
13. Kool, W., Van Hoof, H., Welling, M.: Attention, learn to solve routing problems! arXiv preprint arXiv:1803.08475 (2018)

14. Li, J., Ma, Y., Gao, R., Cao, Z., Lim, A., Song, W., Zhang, J.: Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. IEEE Transactions on Cybernetics **52**(12), 13572–13585 (2021)
15. Liang, H., Wang, S., Li, H., Ye, H., Zhong, Y.: A trade-off algorithm for solving p-center problems with a graph convolutional network. ISPRS International Journal of Geo-Information **11**(5), 270 (2022)
16. Luthans, F., Stajkovic, A.D.: Reinforce for performance: The need to go beyond pay and even rewards. Academy of Management Perspectives **13**(2), 49–57 (1999)
17. Ma, Y., Li, J., Cao, Z., Song, W., Zhang, L., Chen, Z., Tang, J.: Learning to iteratively solve routing problems with dual-aspect collaborative transformer. Advances in Neural Information Processing Systems **34**, 11096–11107 (2021)
18. McKenzie, G.: Spatiotemporal comparative analysis of scooter-share and bike-share usage patterns in washington, dc. Journal of transport geography **78**, 19–28 (2019)
19. Nie, S., Cai, G., He, J., Wang, S., Bai, R., Chen, X., Wang, W., Zhou, Z.: Economic costs and environmental benefits of deploying ccus supply chains at scale: Insights from the source–sink matching lca–milp approach. Fuel **344**, 128047 (2023)
20. Sotiropoulou, K.F., Vavatsikos, A.P.: A decision-making framework for spatial multicriteria suitability analysis using promethee ii and k nearest neighbor machine learning models. Journal of Geovisualization and Spatial Analysis **7**(2), 20 (2023)
21. Vahidnia, M.H., Vahidi, H., Hassanabad, M.G., Shafiei, M.: A spatial decision support system based on a hybrid ahp and topsis method for fire station site selection. Journal of Geovisualization and Spatial Analysis **6**(2), 30 (2022)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)