Report on Solving Probabilistic Maximal Covering Location–Allocation Problem using Artificial Bee Colony Algorithm with Local Refinement

Subhrajit Das
Supervisors:
Dr. Priya Ranjan Sinha Mahapatra
Dr. Soumen Atta
Department of Computer Science and Engineering
University of Kalyani
iamsubhrajit10@gmail.com

March, 2023

***Abstract*—This report proposes a solution to the NP-Hard Probabilistic Maximal Covering Location-Allocation Problem (PMCLAP) using the Artificial Bee Colony algorithm with regional facility updation (local refinement). The approach is aimed at improving the results by applying the regional facility updation procedure for quick convergence. For attaining optimal benchmark results regarding demand and time required for solving the allocation problem, several strategies are suggested. The effectiveness of the proposed solving strategy is discussed, and its capability in addressing complex PMCLAPs is highlighted. The approach has been tested on three datasets of different sizes: 30 instances, 324 instances, and 818 instances. While good results were obtained for the smaller dataset, the larger dataset instances of 324 and 818 were found to be more challenging.**

KEYWORDS

Facility location-allocation problem, Covering location allocation problem, Probabilistic maximal covering location-allocation problem (PMCLAP), Artificial bee colony algorithm (ABC), Local refinement

CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

Introduced by Marianov and Serra [1998], the probabilistic maximal covering location-allocation problem (PMCLAP) is a renowned NP-hard problem in operations research which is a modified and constraints imposed on the maximal covering location problem (MCLP) proposed by Church and ReVelle [1974]. The application of this problem is enormous for example locating places for first aid centers, hospitals, fire stations,locating electric vehicle (EV) charging stations, stores of fast food chains, placing ATMs, etc.

In recent years, swarm intelligence-based algorithms have shown great potential in solving optimization problems. One of such popular swarm intelligence-based algorithms is the Artificial Bee Colony algorithm (ABC) introduced by Karaboga et al. [2005]. As the PMCLAP problem is also an optimization problem, we have proposed to use the ABC algorithm to solve PMCLAP problem, which incorporates a regional facility updation strategy (Mukhopadhyay et al. [2015];Jain et al. [1999];) for quick convergence. The swarm-based optimization algorithm Artificial Bee Colony (ABC) is inspired by honeybees foraging behavior. The algorithm involves three types of bees: employed bees, onlooker bees, and scout bees. Employed bees perform local search around their current solutions and communicate their findings to onlooker bees, who use this information to select promising solutions. Scout bees explore the search space for new solutions. These three bees procedure is considered as one cycle of iteration. An iteration refers to one complete cycle of the algorithm, which involves generating and evaluating candidate solutions, selecting the best solutions, and updating the search process based on the information gathered from the previous iterations. The number of iterations in ABC is typically specified as a stopping criterion and can be adjusted based on the problem complexity and required solution accuracy. Algorithm 1 shows the basic process of ABC Algorithm Karaboga [2010] In recent years, swarm intelligence-based algorithms have shown great potential in solving optimization problems. One of such popular swarm intelligence-based algorithms is the Artificial Bee Colony algorithm (ABC) proposed by Karaboga [2010], Karaboga et al. [2005]. As the PMCLAP problem is also an optimization problem, we have proposed to use the ABC algorithm to solve PMCLAP problem, which incorporates a regional facility updation strategy (Mukhopadhyay et al. [2015];Jain et al. [1999];) for quick convergence. The swarm-based optimization algorithm artificial bee colony (ABC) is inspired by honeybees foraging behavior. Three kinds of bees are involved in this algorithm: employed bees, onlooker bees, and scout bees. Employed bees search locally around their current solutions and communicate their findings to onlooker bees, who use this information to select promising solutions. Scout bees explore the search space for new solutions. These three bees procedure is considered as one cycle of iteration. An iteration refers to one complete cycle of the algorithm, which involves generating and evaluating candidate solutions, selecting the best solutions, and updating the search process based on the information gathered from the previous iterations. The number of iterations in ABC is typically specified as a stopping criterion and can be adjusted based on the problem complexity and required solution accuracy. Algorithm 1 shows the basic process of ABC Algorithm Karaboga [2010]

Algorithm 1 Artificial Bee Colony algorithm

```
procedure ABC
    Initialization Phase
    repeat
        Employed Bees Phase
        Onlooker Bees Phase
        Scout Bees Phase
        Memorize the best solution achieved so far
    until (Cycle = Maximum Cycle Number or a
  Maximum CPU time or Convergence Criteria)
end procedure
```

ABC algorithms are known for solving various NP-hard problems with near-global optimal result, but they can have a high computation time. Local improvement strategies can be incorporated into the ABC algorithm to improve its performance. In this approach, candidate solutions are encoded for probable facility locations, and the nectar or objective function is calculated as the total demand covered. A regional facility updation strategy is considered to fine-tune the food sources and improve convergence speed. Comparison between the proposed ABC-algorithm technique with regional facility updation and the ALNS method is done with respect to computational time and total demand covered. Experimental results show that the suggested technique outperforms ALNS in most cases in terms of computational time, but it still lacks in covering the total demand served.

The report is arranged as follows: similar works are discussed in section II; section III mathematically defines PMCLAP; Section IV describes the suggested ABC technique in detail; section V reports and discusses the experimental results; and at the end report is concluded in section VI .

II. Literature Review

From the introduction of PMCLAP by Marianov and Serra [1998], various similar versions have been introduced in probabilistic location allocation literature (Marianov and Serra [1998];de Assis Corrêa et al. [2007];de Assis Corrêa et al. [2009];Pereira et al. [2015]). The MCLP, introduced by Church and ReVelle [1974], aims at locating p facilities among n possible candidates, such that the maximal possible population is served. However, this problem does not consider the capacities or congestion issues. To overcome this limitation, Marianov and Serra [1998] introduced the

PMCLAP, an extension of the MCLP in which a minimum quality on the service level is imposed, assuming that clients arrive to the facilities accoridng to a Poisson distribution. The way to measure the demand at a facility is either by counting the number of people waiting for service, or by measuring the waiting for the service. In the present work, we consider the model of PMCLAP proposed by Marianov and Serra [1998]. Several researchers have proposed various methods to solve the PMCLAP includes Hybrid Heuristics algorithm, which combines both the Genetic Algorithm (GA) and the Simulated Annealing (SA) methods de Assis Corrêa et al. [2007].de Assis Corrêa et al. [2009] proposed a decomposition approach for the PMCLAP. The proposed method decomposes the original problem into a set of subproblems, each of which is then solved using a GA-based algorithm. The solution of the original problem is obtained by combining the solutions of the subproblems.The authors of Pereira et al. [2015] proposed a hybrid method for solving the probabilistic maximal covering location-allocation problem (PMCLAP). The proposed method combines simulated annealing (SA) and a genetic algorithm (GA) to improve the quality of the solutions. The SA method is used to generate initial solutions, while the GA method is used to refine the solutions obtained by SA. The proposed hybrid method was tested on benchmark instances and compared with other existing methods, and the results showed that the hybrid method outperforms the other methods in terms of solution quality and computational time. Huang et al. [2022] proposed a particle swarm optimization (PSO) algorithm for solving the PMCLAP. The proposed method uses a population of particles to search for the optimal solution. The algorithm works by updating the velocity and position of the particles based on the best solution found so far.

III. Problem definition

Formally, the problem is defined on a graph with a set N of n nodes. Each node is associated with a demand d_i and a service radius (or covering distance) S_i in case a facility is located at the facility candidate i. Without loss of generality, a service radius equal to S is considered for all the facilities. Let N_i be a subset containing the list of nodes within S units of distance from node i, i.e., the set of location candidates j that can serve client i. In the PMCLAP, f_i represents the contribution of client i to the system congestion. This value is calculated as a fraction of the client's demand. It is assumed that clients will arrive to a facility according to a Poisson distribution with parameter rate μ. The parameter α defines the minimum probability of at most

- a queue with b clients, or;
- a waiting time of τ minutes.

To model the PMCLAP, we define two sets of binary variables: one of location and the other for one for allocation decisions. Variables y_j are equal to one if and only if location j $\in$ N is opened, and variables x_{ij} are equal to one if and only if the demand of node i is associated with facility j, i,j $\in$ N. The problem can be formulated as follows:

$$maximize \sum_{i \in N} \sum_{j \in N_i} d_i x_{ij} \tag{1}$$

subject to

$$\sum_{j \in N_i} x_{ij} \leq 1, \quad i \in N \tag{2}$$

$$\sum_{i \in N} y_j = p \tag{3}$$

$$x_{ij} \leq y_j, \quad i \in N,\ j \in N \tag{4}$$

$$\sum_{i \in N} f_i x_{ij} \leq \mu \sqrt[b+2]{1-\alpha} \quad j \in N \tag{5}$$

$$\sum_{i \in N} f_i x_{ij} \leq \mu + \frac{1}{\tau} \ln(1-\alpha), \quad j \in N \tag{6}$$

$$y_j, x_{ij} \in \{0,1\}, i \in N, j \in N \tag{7}$$

The optimization problem aims to maximize the total served demand, with constraints ensuring the allocation and location of facilities. Specifically, objective function (1) maximizes the total served demand, while constraints (2) guarantee that each client is served by at most one facility, and constraint (3) determines the number of facilities to be opened. Constraints (4) link the location to allocation variables, allowing clients to be allocated to an opened facility. Constraints (5) ensure that facility j has fewer than b clients in the queue with at least probability α, while constraints (6) ensure that the waiting time for service at facility j is at most τ minutes with a probability of at least α. Binary nature of the variables is enforced by constraints (7). It is worth noting that x_{ij} equals zero for all j $\notin$ N_i.

Constraints (5) and (6) account for the probabilistic nature of the problem. Following the approach of Marianov and Serra [1998], an M/M/1/∞/FIFO queueing system is assumed, where service requests occur at each demand node i according to a Poisson process with intensity f_i. As customers arrive at a facility j from different demand nodes, the request for service at this facility is the sum of several Poisson processes with an intensity of λ_j, calculated as:

$$\lambda_j = \sum_{i \in N} f_i x_{ij} \tag{8}$$

IV. Proposed ABC for PMCLAP

In this section, the proposed ABC-based solution for the aforementioned PMCLAP is described. The overall procedure of the proposed ABC-based solution of PMCLAP is demonstrated in Fig. 1. In the following subsections, we describe each step of the proposed method in detail.

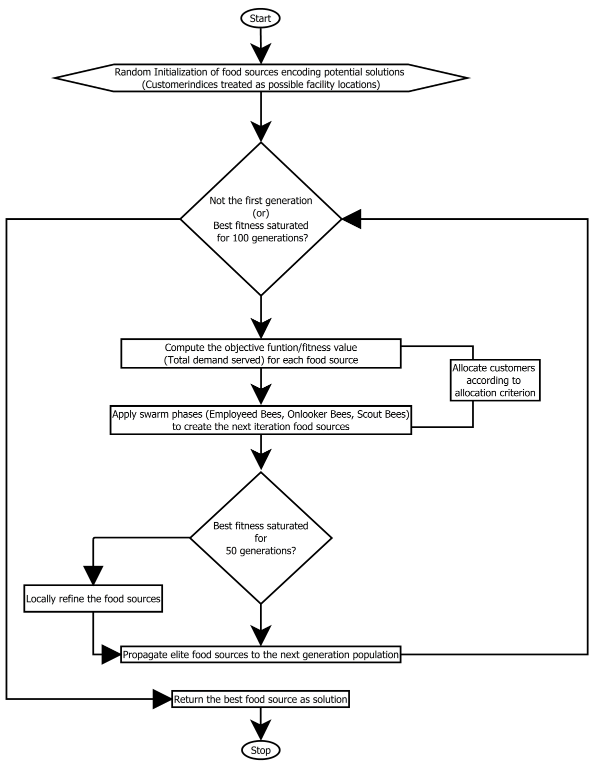

Fig. 1. Flowchart of the proposed ABC-based method for solving PMCLAP

A. Solution encoding

In the ABC algorithm, solutions to the optimization problem are encoded in a similar way to the chromosome encoding in genetic algorithms Atta et al. [2018]. For the PMCLAP, a possible solution with k facilities is a set of potential locations of k facilities chosen from the set of m customers, P = {p_1, p_2, ..., p_m}. Therefore, a food source encodes an integer string {t_1, t_2, ..., t_k} of length k representing the indices of the k customers selected as the facilities. Each element, $p_{ti} \in$ P for i={1, 2, ..., k}, since the potential locations of the k facilities are limited to the locations of the customers themselves.

B. Solution Vector (Food Sources) initialization

Similar to Atta et al. [2018] the initial population consists of ρ solutions where ρ is a user defined parameter called solution vector size. Each food source of the initial solution vector is generated by selecting k random indices from the set {1,2,...,m}. In this work, we set ρ=30 which is chosen experimentally.

C. Objective function computation

Objective function of a solution/food source represents the goodness of the solution encoded in it with respect to PMCLAP. The objective of PMCLAP is to maximize the coverage (i.e., the total demands of the customers covered by some facilities satisfying at least the minimum service quality). Hence coverage of the solution encoded in a food source is considered as the objective function of the food source, as shown in Eq. 1. The objective function is to be maximized.

D. Allocation of customers

Multiple facilities may be available within the service radius of a particular customer, and selecting the appropriate facility is crucial for achieving an optimal solution. Several allocation strategies have been proposed by researchers, including hybrid approaches de Assis Corrêa et al. [2007]. To achieve faster customer allocation, we employed three strategies interchangeably to obtain better experimental results: allocating the customer to the nearest feasible facility de Assis Corrêa et al. [2007], allocating the customer to the least congested feasible facility de Assis Corrêa et al. [2009], and allocating the customer according to the feasible facility with the maximum weighted demand.

E. Swarm in ABC Algorithm

1) Employed Bees

As Karaboga [2010] mentioned, employed bees search for new food sources having more nectar within the neighbourhood of the food source in their memory. They find a neighbour food source and then evaluate its profitability (fitness). Here, greedy strategy is used, for every food source in memory, one random neighbour is picked up and if the chosen neighbour has more nectar then it is taken into food sources otherwise we go with the original food source.

2) Onlooker Bees

In the ABC algorithm, onlooker bees choose a food source probabilistically based on the information shared by employed bees Karaboga [2010]. This is done using a fitness-based selection method such as the roulette wheel selection method Goldberg [1989]. Once a food source is chosen, the onlooker bee determines a neighbourhood source and computes its fitness value. Greedy selection is then applied, such that more onlookers are recruited to richer sources, leading to positive feedback behavior.

3) Scout Bees

The scouts in the ABC algorithm refer to the unemployed bees that randomly choose their food sources Karaboga [2010]. If an employed bee's solution cannot be improved through a predetermined number of trials, which is specified by the user and referred to as the "limit" or "abandonment criteria", the employed bee becomes a scout and abandons its solution. The abandoned solution is then randomly searched by the scout to find a new solution. In this study, a limit value of 50 was chosen. Therefore, poor sources, whether initially or due to exploitation, are abandoned, which creates a negative feedback behavior to balance the positive feedback.

4) Local refinement

After the scout bees phase, each food source (solution) is locally updated. Local improvement is done as follows Atta et al. [2018]. First, the customers are clustered with respect to the facilities, i.e., each customer p_i is assigned to its nearest facility c_j. A cluster $clst_j$ is formed around each facility at c_j with customers assigned to it (Jain et al. [1999]; Mukhopadhyay et al. [2015]). Thereafter, each facility at c_j encoded in a food source is updated with c_t such that:

$$t = \arg \min_{p_i \in clst_j} \sum_{p_l \in clst_j} f_l \cdot d(p_i, p_l) \tag{9}$$

This ensures that the facilities' locations are shifted towards more centrally located points with higher demand values, which increases the overall coverage quickly (Atta et al. [2018]). The refinement strategy also tries to provide better service to customers near the facility compared to those far away from the facility. However, once the best solution does not change for 50 consecutive iterations, the local refinement procedure is not applied anymore to allow the food sources to evolve freely.

5) Memorize best solutions

To prevent loss of promising solutions resulting from the stochastic nature of the ABC swarm phases, it is necessary to store the best solutions obtained up to the current iteration. To achieve this, the solution vectors of the current iteration are compared with those of the previous iteration, and a greedy strategy is employed to select the solution with the higher nectar. This comparison is carried out sequentially for each solution vector in the iteration. This approach ensures that the best food source found thus far is retained.

6) Termination criterion

The iterative process of nectar computation involves employed bees, onlooker bees, scout bees, local refinement, and memorization of the best solutions across multiple generations. Local improvements of the food sources are executed until the best fitness value remains unchanged for 50 generations. After this, local refinement is discontinued and the food sources are allowed to evolve freely. The iterative process continues until the best fitness value remains unchanged for the last 100 iterations. The output of ABC is determined by the best solution, which corresponds to the highest total demand served.

V. EXPERIMENTAL RESULTS

This section provides details and results of the computational experiments carried out to asses the quality of the ABC with local refinement. The algorithm was coded in Matlab™ version: R2021a. All computations were performed on machines equipped with an Intel i5™ processor running at 2.5 GHz. The proposed algorithm needed less than 500 MB of RAM memory.

TABLE I
COMPUTATIONAL RESULTS FOR THE INSTANCE SET WITH 30 NODES USING LEAST CONGESTED ALLOCATION STRATEGY

Instance	ANLS			ABC with local refinement				
	Best	Average	Time(s)	Best	Average	Std Dev	Gap (%)	Time(s)
30_2_0_0_85	3700	3700	12.333	3700	3700	0	0	1.71
30_2_0_1_85	5100	5100	9.000	5090	5078	12	0.19	1.33
30_2_0_2_85	5210	5210	4.667	5210	5210	0	0	1.38
30_2_0_2_95	4520	4520	11.333	4520	4520	0	0	1.34
30_3_0_0_85	5390	5390	5.000	5390	5378	12	0	1.51
30_3_0_1_85	5390	5390	3.000	5390	5390	0	0	1.53
30_3_0_1_95	5270	5270	44.667	5240	5240	0	0.56	1.66
30_3_0_2_85	5390	5390	3.667	5390	5390	0	0	1.41
30_3_0_2_95	5390	5390	3.667	5390	5378	12	0	1.50
30_4_0_1_95	5390	5390	8.333	5390	5390	0	0	1.83
30_5_0_0_95	5330	5330	288.667	5300	5274	11.66	0.56	1.81
30_6_0_0_95	5410	5410	62.333	5390	5390	0	0.36	1.98

Similar to the approach in Pereira et al. [2015], we solved a benchmark dataset of instances consisting of three classes: 30 nodes, 324 nodes, and 818 nodes. Each instance was solved thirty times with the number of facilities to open (p) varying from two to 50. The service radius was set to 1.5 miles for the instances with 30 nodes, 250 m for instances with 324 nodes, and 750 m for instances with 818 nodes. The 30-node dataset was originally proposed by Marianov and Serra Marianov and Serra [1998], while the 324 and 818-node datasets were proposed by de Assis Corrêa et al. [2007].

The tables I, V, and V provide the names of the instances along with the values of parameters used for each run. The name of an instance, such as 30_2_0_0_85, indicates that it represents a 30-node problem with two facilities, where the congestion type is based on the number of clients (0 for queue size, 1 for waiting time), the congestion parameter is either the number of clients b on the queue or the waiting time τ in minutes, and the minimum probability α is given as a percentage value. For the 30-node network, the rate parameter μ is fixed at 72, while for the 324- and 818-node networks, it is fixed at 96. The parameter f_i that appears in formulations (1)-(7) is calculated as fd_i, where f is 0.01 for the 324- and 818-node networks, and either 0.015 (for queue size type constraints) or 0.006 (for waiting time type constraints) for the 30-node network.

Tables I-III display the best and average solutions (if available), computation time, and standard deviation for the ABC algorithm with local refinement for each instance. The study found that for the 30 node dataset, the strategy proposed by de Assis Corrêa et al. [2009] - allocating to the least congested facility - achieved better results with ABC almost reaching the benchmark. For the 324 and 818 node datasets, allocating according to the weighted demand strategy led to better results, although neither dataset met the benchmark. The Gap in % shows the difference in percentage between the obtained and benchmark results. Although not achieving the benchmark, the ABC algorithm had significantly faster convergence than ANLS Pereira et al. [2015].

TABLE II
COMPUTATIONAL RESULTS FOR THE INSTANCE SET WITH 324 NODES USING MAXIMUM WEIGHTED DEMAND FACILITY ALLOCATION STRATEGY

	ANLS		ABC with local refinement				
Instance	Best	Average	Best	Average	Standard Deviation	Gap (%)	Time(s)
324_10_0_0_85	37180	37180.0	36800	36441.80	167.68	1.02	97.40
324_10_0_0_95	21460	21460	21125	20912.63	103.38	1.56	80.28
324_10_0_1_85	51000	51000	49939	49272.63	205.59	2.08	70.25
324_10_0_1_95	35360	35360	35120	34972.73	95.62	0.67	62.60
324_10_0_2_85	59740	59740	58598	58033.50	202.97	1.92	66.05
324_20_0_0_85	74360	74357.7	72181	71382.56	252.31	2.93	68.21
324_20_0_0_95	42920	42919.7	40997	40641.73	206.35	4.48	71.00
324_20_0_1_85	101978	101974.0	95975	95306.56	338.52	2.94	75.58
324_20_0_1_95	70720	70720	69219	68685.73	233.16	2.12	64.49
324_20_0_2_85	119455	119447.3	113586	111828.70	1070.86	4.91	87.94
324_20_0_2_95	90780	90780	87788	86898.23	573.35	3.29	71.18

TABLE III
COMPUTATIONAL RESULTS FOR THE INSTANCE SET WITH 818 NODES USING MAXIMUM WEIGHTED DEMAND FACILITY ALLOCATION STRATEGY

	ANLS		ABC with local refinement				
Instance	Best	Average	Time(s)	Best	Average	Gap (%)	Time(s)
818_10_0_0_95	21460	21460	6818	21412	21371.90	0.22	109.84
818_10_0_1_95	35360	35360	10407	35326	35231.66	0.09	115.36
818_10_0_2_95	45390	45390	7400	45318	45219.53	0.15	162.61
818_20_0_0_85	74360	74360	17183.66	73278	73037.80	1.45	45.39
818_20_0_0_95	42920	42920	9890.33	42661	42422.80	0.60	36.96
818_20_0_1_85	102000	102000	16602.00	101010	100547.20	0.97	35.10
818_20_0_1_95	70720	70720	16880.66	70144	69969.40	0.81	36.40
818_20_0_2_85	119480	119480	19074.66	118336	118065.40	0.95	59.38
818_20_0_2_95	90780	90780	14622.66	89950	89719.30	0.91	54.99
818_50_0_0_85	185637	185587.7	24094.33	180641	180317.30	2.69	116.72
818_50_0_1_85	254996	254987.3	23978.00	249494	248645.40	2.15	129.97
818_50_0_2_85	298692	298648.3	23729.66	291707	291341.60	2.33	118.63

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisors, Dr. Priya Ranjan Sinha Mahapatra from University of Kalyani, Department of Computer Science and Engineering, and Dr. Soumen Atta from University of Nova Gorica, School of Engineering and Management, for their guidance, support, and encouragement throughout my research. Their insights and expertise have been invaluable in shaping my work and pushing me to new heights.

I would also like to thank the Department of Computer Science and Engineering at the University of Kalyani for providing access to the lab resources that were crucial to the success of this project. The moral support and encouragement from the faculty and staff in the department have been a constant source of inspiration.

Finally, I would like to express my deep appreciation to my family and friends who have supported me throughout my academic journey. Their love, encouragement, and unwavering support have been the foundation of my success.

VI. CONCLUSION

In conclusion, this study proposed an approach to solve the Probabilistic Maximal Coverage Location-Allocation Problem using the Artificial Bee Colony algorithm with local refinement strategy. Three allocation subproblems were solved using different strategies, resulting in promising results for the 30 node dataset. However, for larger datasets of 324 and 818 nodes, the benchmark results were missed by a small margin. Future work will focus on making appropriate changes to achieve benchmark results while maintaining faster computational time. Overall, this study provides insights into solving the PMCLAP problem and opens up avenues for further research in this area.

REFERENCES

Soumen Atta, Priya Ranjan Sinha Mahapatra, and Anirban Mukhopadhyay. Solving maximal covering location problem using genetic algorithm with local refinement. *Soft Computing*, 22:3891–3906, 2018.

Richard Church and Charles ReVelle. The maximal covering location problem. In *Papers of the regional science association*, volume 32, pages 101–118. Springer-Verlag Berlin/Heidelberg, 1974.

Francisco de Assis Corrêa, Antonio Augusto Chaves, and Luiz Antonio Nogueira Lorena. Hybrid heuristics for the probabilistic maximal covering location-allocation problem. *Operational Research*, 7:323–343, 2007.

Francisco de Assis Corrêa, Luiz Antonio Nogueira Lorena, and Glaydston Mattos Ribeiro. A decomposition approach for the probabilistic maximal covering location-allocation problem. *Computers & Operations Research*, 36(10):2729–2739, 2009.

David E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

Ying-Ying Huang, Quan-Ke Pan, Liang Gao, Zhong-Hua Miao, and Chen Peng. A two-phase evolutionary algorithm for multi-objective distributed assembly permutation flowshop scheduling problem. *Swarm and Evolutionary Computation*, 74:101128, 2022.

Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

Dervis Karaboga. Artificial bee colony algorithm. *scholarpedia*, 5(3):6915, 2010.

Dervis Karaboga et al. An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer ..., 2005.

Vladimir Marianov and Daniel Serra. Probabilistic, maximal covering location—allocation models forcongested systems. *Journal of Regional Science*, 38(3): 401–424, 1998.

Anirban Mukhopadhyay, Ujjwal Maulik, and Sanghamitra Bandyopadhyay. A survey of multiobjective evolutionary clustering. *ACM Computing Surveys (CSUR)*, 47(4):1–46, 2015.

Marcos A Pereira, Leandro C Coelho, Luiz AN Lorena, and Ligia C De Souza. A hybrid method for the probabilistic maximal covering location–allocation problem. *Computers & Operations Research*, 57:51–59, 2015.