



A solution method to maximal covering location problem based on chemical reaction optimization (CRO) algorithm

Md. Shymon Islam¹ · Md. Rafiqul Islam¹

Accepted: 27 February 2023 / Published online: 20 March 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

The maximal covering location problem refers to the problem of finding an optimal placement of given number of facilities to a network. The objective is to maximize the total demands of the covered population within some constraints. Several metaheuristic approaches were proposed to solve the problem as it is an NP hard problem. In this article, we have proposed a chemical reaction optimization (CRO)-based approach to solve MCLP. CRO is a metaheuristic based on population to solve optimization problems. We are proposing a method to solve MCLP by redesigning four fundamental operators of CRO. Sometimes the solutions get trapped into local maxima, so an additional repair operator is also designed to find optimal solutions. The proposed algorithm is tested for both small and large scales of instances of datasets, which include benchmark as well as random ones. The proposed method gives best percentage of coverage results in 91.60% of instances, and for the remaining 8.40% of instances it produces results with average error value 0.10% which is very close to the optimal value. Nevertheless, the proposed method performs very well in terms of computational time for all test instances (100%) on all datasets compared to state-of-the-art method (Atta_GA). Wilcoxon signed-rank test has been performed on the results of the proposed method to observe the statistical significance. For both real-world and random instances, the results of the statistical test are significant.

Keywords Chemical reaction optimization · Maximal covering location problem (MCLP) · Molecule · NP hard · Synthesis

1 Introduction

The covering location problem (CLP) is a well-known problem discussed in location science. There are two types of CLPs. They are the set covering location problem (SCLP) and the maximal covering location problem (MCLP) (Zarandi et al. 2013). MCLP deals with the problem of finding an optimal placement of a given fixed number of facilities on a network. The goal of MCLP is to determine an optimal placement in such a way that the total demands covered by the served population is maximized. If a customer falls within a given constant service area or coverage area, then the node is served by a facility (Atta et al. 2018). MCLP is a resource constraints problem. The objective of the problem is to serve the

demands of customers as much as possible with the limited resources or budget (Davari et al. 2011). Maximal covering location problem is also a constraint satisfaction problem that is very useful in locating objective areas to be served in different contexts.

There are many applications where MCLP can be applied for better solutions such as identifying police patrol stations, rehabilitating ambulances, ensuring first aid privileges during natural disasters and providing emergency services in rural as well as over populated areas (Zarandi et al. 2013). Dynamic MCLP has recently been applied for locating fire station and allocating facilities during wars and natural disasters (Hajipour et al. 2022). In emergency and military services, the MCLP has several applications to deploy (Atta et al. 2018). The MCLP has importance not only in private sectors but also in public sectors. Plant and warehouse placements and the placement of telecommunication antennas, etc., are examples of several private sectors where MCLP can be applied to establish an optimal placement structure. Similarly, the placement of schools, libraries, parks, bus stops, hospitals, fire stations and emergency units are common

✉ Md. Shymon Islam
shymum1702@cseku.ac.bd

Md. Rafiqul Islam
dmri1978@gmail.com

¹ Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

examples of public services where MCLP is applicable to determine optimal solution (Lorena and Pereira 2002). For sharing economy, MCLP can be applied for better solutions. MCLP has recently been deployed to distribute bikes in Chinese cities (Li et al. 2020). It is greatly applicable in designing network with constraints.

There are several strategies to solve the maximal covering location problem. It is an NP-hard problem, so heuristics and metaheuristics approaches are suitable for this context. Many heuristics and metaheuristics algorithms were proposed to solve MCLP by the researchers such as GA (genetic algorithm)(Atta et al. 2018; Zarandi et al. 2011), SA (simulated annealing)(Zarandi et al. 2013; Davari et al. 2011), GH (greedy heuristics)(Berman and Krass 2002; Máximo et al. 2017), TS (tabu search)(Bagherinejad et al. 2018). In our study, we have chosen chemical reaction optimization (CRO) algorithm, a nature inspired metaheuristics, to deal with MCLP. There are some reasons to select CRO over other metaheuristics approaches. Firstly, CRO can serve the purposes of both GA and SA algorithms. GA is a Darwinian evolutionary algorithm, functions on the basis of biologically inspired operators selection, crossover and mutation (Atta et al. 2018) where mutation and crossover operators are similar to CRO's synthesis and decomposition. SA is another well-known stochastic metaheuristic; the energy conservation mechanisms of CRO provide a homogeneous effect like SA (Lam and Li 2012). So, CRO can provide a dual effect of GA and SA algorithms at the same time. Secondly, the best feature of CRO is its ability to search. It can search the solution space locally as well as globally with the help of its four operators (on-wall ineffective collision, intermolecular ineffective collision, decomposition and synthesis) (Lam and Li 2012). It makes the algorithm more efficient compared to other metaheuristics approaches. Thirdly, additional operators are supported in CRO to make a possible solution on the problem-specific arisen issue. In our study to deal with MCLP, we have faced the problem of solutions are sometimes being trapped into local maxima, so an additional repair operator is designed to resolve the arisen problem. Fourthly, the dynamic behavior of CRO can increase or decrease the number of solutions according to the necessity of current iteration to reach in an equilibrium energy conservation state which is quite impressive. So, the number of solutions varies from iteration to iteration. Fifthly, in recent years, CRO has successfully solved many optimization problems such as determining the machining parameter of abrasive water jet cutting (Bhoi et al. 2022), designing reduced congestion road networks (Salman and Alaswad 2022), mobile robot path planning with obstacle avoidance (Islam et al. 2021), longest common subsequence problem for multiple string (Islam et al. 2019), DNA motif discovery (Saha et al. 2020), 0–1 knapsack problem (Truong et al. 2013), task scheduling and grid computing problem (Jin et al. 2011), quadratic

assignment problem (Xu et al. 2010), the max flow problem (Barham et al. 2016), shortest common super-sequence problem (Saifullah and Islam 2016), RNA structure prediction (Kabir and Islam 2019), transportation scheduling in supply chain management (Mahmud et al. 2017), RNA secondary structure prediction (Islam et al. 2021), optimization of protein folding (Islam et al. 2020), flexible job-shop scheduling problems with maintenance activity (Li and Pan 2012), the distributed permutation flow-shop scheduling problem with makespan criterion (Bargaoui et al. 2017), the cloud job scheduling (Zain and Yousif 2020), etc., with better results than the other existing metaheuristic algorithms.

Different researchers used different optimization techniques for different problems and sub-problems. For the task of codon-mRNA prediction, the hybrid method DLSTM-DSN-WOA was introduced in Kadhuim and Janabi (2023) where whale optimization algorithm (WOA) is used to build optimal structure for deep long short-term memory (LSTM). For the task of model parameter estimation of the proton exchange membrane fuel cell (Syah et al. 2023), the improved teamwork optimizer is used for optimization. Another optimization problem, verifying chemical reaction (VCR) was solved using whale optimization algorithm in Janabi et al. (2023). Whereas in this study, we have solved the problem termed as maximal covering location problem (MCLP) using the optimization algorithm named chemical reaction optimization (CRO) algorithm.

In this article, we have proposed a CRO-based approach to solve MCLP. The molecules of CRO are encoded by locating suitable facilities, and the potential energy is computed with respect to percentage of coverage and computational time. To measure the performance of the proposed method, we have tested five real-world datasets as well as four randomly generated datasets consisting of 324 to 2500 nodes (including small, medium, and large datasets). The obtained results of the proposed method are compared with the GA-based method proposed by Atta et al. (2018) which is the state-of-the-art method. The authors of paper (Atta et al. 2018) compared their method with two other methods (Davari et al. 2011) and (Lorena and Pereira 2002). The proposed method obtained better results in almost all the test instances in terms of percentage of coverage as well as computational time compared to Atta_GA (Atta et al. 2018). During dealing with MCLP, a problem is noticed that some solutions are getting trapped into local optima, so an additional repair operator (LM-GM) is designed to resolve the trapping situation; all the results are examined and shown with and without LM-GM to observe the improvements in this context. The novelty and contributions of the proposed work are as follows:

1. We have redesigned four fundamental operators of CRO algorithm and solved the MCLP, and this approach is not used before to solve the MCLP.

2. A repair operator has been designed to overcome the trapping into local optima, and it produces the max point in MCLP that helps to obtain the best results in less computational time.
3. We have obtained best known results in almost all the cases of the five real-world datasets. The proposed algorithm gives a substantial improvement in computational time.
4. The proposed algorithm gives the best results in comparison with the state-of-the-art algorithm with less computational time in four random datasets.
5. A statistical test has been examined that shows the proposed algorithm is significant in both real-world and random instances.

In this article, so far we have demonstrated the basic ideas of MCLP in Sect. 1. The problem statement and objective function are briefly described in section 2. Section 3 is for related work; Sect. 4 demonstrates our proposed method for solving MCLP using chemical reaction optimization, Sect. 5 describes the experimental results and analysis of the work, and Sect. 6 concludes the work.

2 Problem statement and objective function

The MCLP deals with the problem of finding an optimal placement of a given fixed number of facilities on a network in such a way that the total demands of the attended population are maximized (Zarandi et al. 2011).

2.1 The hypothesis and limitations of the develop method

Some assumptions have been considered for this problem. MCLP works with a given number of customers (nodes) in a network. Each customer (node) has three basic features. These include the demand value (must be non-negative), x coordinate value and y coordinate value. Two coordinate values (x, y) define a customer's location in the network. Each customer (node) is assigned a fixed number of facilities. Each facility has a service area called coverage area which is circular in shape, and it remains constant throughout the whole processing. A facility can be provided to a customer (node) if the Euclidean distance between the concerned node and the facility node is smaller than the constant service area (coverage area). Therefore, the target of the problem is to locate an optimal placement of the facilities on the network so that the total demands of the served population is maximized. The mathematical demonstration of the objective function of the maximal covering location problem is given by the equation

as follows:

$$f(D) = \text{MAX} \sum_{x \in D} d_x s_x \quad (1)$$

Subject to the constraints:

$$s_x \leq \sum_{y \in E_x} p_y, x \in D \quad (2)$$

$$\sum_{y \in F} p_y = f \quad (3)$$

$$s_x \in \{0, 1\}, x \in D \quad (4)$$

$$p_y \in \{0, 1\}, y \in F \quad (5)$$

Here, formulation Eqs. (1), (2), (3), (4) and (5) are taken from Atta et al. (2018). The goal of the objective function Eq. (1) is to maximize the total demands covered by the selected customers (population). Constraint Eq. (2) shows that a customer is covered when there exists at least one facility within the coverage area of the node. Constraint Eq. (3) denotes that the total number of facilities should be exactly f . Constraints Eqs. (4) and (5) are the two binary decision variables for MCLP. In this article, we did not consider the variable length service distances for different nodes.

2.2 Complexity of MCLP

The maximal covering location problem is NP-hard (Megiddo et al. 1983). The solution of MCLP is non-deterministic in polynomial time as it has to maintain a huge graph or matrix for a small size of customers (nodes), and the total subsets of the customers are also huge in number and the problem is restricted with several constraints. For example, if we consider 500 customers each of which to be placed with exactly 6 facilities, then we need to maintain a distance matrix of 500×500 . To determine the optimal solution from this large solution space, we need to explore a large number of different combinations. For our example, we have total $\frac{500 p_6}{6!}$ number of solutions available. It is necessary to find the best solution (one) from them.

Table 1 describes the complexity of MCLP. The feasible solutions of MCLP are increasing to a great extent with the

Table 1 Complexity of MCLP

No. of customers	No. of facilities	Dimension of distance matrix	Total feasible solutions
100	6	100×100	1.19×9
500	6	500×500	2.11×13
1000	8	1000×1000	2.41×19
2500	10	2500×2500	2.58×27

Table 2 Symbolic notations of the objective function

Symbol	Descriptions
m	The number of all customers (nodes)
N	The set of customers (nodes)
D	The set of demands
F	The set of potential facility sites
x	The index of demand nodes
y	The index of eligible facility sites
d_x	The demand value of customer (node) x
l_{xy}	The distance from node x to facility at node y
r	The constant service radius
f	The number of all facilities to be installed
E_x	All facility sites of node x within r
s_x	It becomes 1 when node x is selected and 0 otherwise
p_y	It becomes 1 when facility y is sited and 0 otherwise

increasing number of customers and facilities. The distance matrix is becoming huge when the number of customers is increasing. MCLP has to maintain several constraints as well. So, there is no deterministic polynomial time solution exists for this problem. That is why metaheuristics approaches are applicable to determine efficient solutions for MCLP. The objective function is basically the fitness function of the problem definition. The notations shown in Table 2 are used to formulate the objective function and constraints for the maximal covering location problem.

Figure 1 depicts a sample example of MCLP. This is a MCLP graph of five customers where each of them asks to be assigned with exactly two facilities. The five customers (nodes) are labeled as $C1$, $C2$, $C3$, $C4$ and $C5$. The corresponding demand value is shown at the top of each node. These values include 50, 4, 33, 15 and 1, respectively. The coordinate values are presented in parentheses. So for customer $C1$ the coordinate values are 409154 and 435528. The service distance and the number of facilities to be installed for this network are 100 and 2, respectively. This is a fully connected MCLP graph. To solve this graph, first create a distance matrix from the coordinate values using Euclidean distance formula. Then, generate all possible solutions or sequences for five customers with 2 facilities. The total number of sequences is $\frac{5P_2}{2!}$ that is ten. Each sequence needs to traverse the entire solution space and find the result. Hence, ten sequences must have ten results, among them the highest one will be the required result. In the sample example, the maximum value is 83, which is generated for sequence 10100. The final percentage of coverage is determined by the average between sum of all customer demand values and the generated result. Here, the sum of all demand values is 103. In Fig. 1, nodes $C1$ and $C3$ are selected as it maximizes the demand values of attended nodes. Hence, $C1$ and $C3$ are

colored. Therefore, the estimated result for the MCLP graph in Fig. 1 is 83. So, the percentage of coverage is $\frac{83}{103} \times 100$ that is 80.58%.

Figure 2 is taken from Zarandi et al. (2011), it represents the visualization to a solution of MCLP with 20 facilities. In the figure the bold nodes are the facilities and the circle denotes the service radius of the corresponding facility. The smaller dots refer to the center of demand that should be served.

3 Related works

Various metaheuristics approaches were proposed to solve the maximal covering location problem. Each approach has some efficiencies as well as pitfalls over other approaches. Some of the existing algorithms are described here.

3.1 Genetic algorithm

In 2018, Atta S et al. proposed an approach to solve the maximal covering location problem (MCLP) using genetic algorithm with local refinement (Atta et al. 2018). Since MCLP is an NP-hard problem, the authors solved the problem using genetic algorithm (GA) that is a standard metaheuristic approach. They created the initial population by encoding chromosomes. For encoding, they used binary array representation strategy. Each chromosome is a solution for GA that can be created randomly or following some preconceived notions. A set of chromosomes makes up the initial population. The authors redesigned the mutation, crossover and selection operators. They also designed a local refinement operator with elitism to get the best results. In each iteration, all chromosomes traverse these operators and try to extract a better solution than the ones that exist. A new generation is created at the end of the current iteration until the desired result is obtained. They used two types of datasets to measure the performance and efficiency of their proposed algorithm. SJC324, SJC402, SJC500, SJC708 and SJC818 are the five real-world datasets that they used. These five datasets are networks of 324, 402, 500, 708 and 818 nodes, respectively. The datasets pmed32 and pmed39 are networks of 700 and 900 nodes, respectively, collected from the OR library. $G\&R100$, $G\&R150$, $ZDS1800$ and $ZDS2500$ are four random datasets used by them to measure both small- and large-scale variation of MCLP. These four datasets are networks of 100, 150, 1800 and 2500 nodes, respectively. They compared their proposed algorithm with other approaches in terms of percentage of coverage and computational time. In most cases, their proposed method gives quite good results in both percentage of coverage and computational time.

Their proposed method is almost an optimal and efficient approach that can solve both small- and large-scale versions

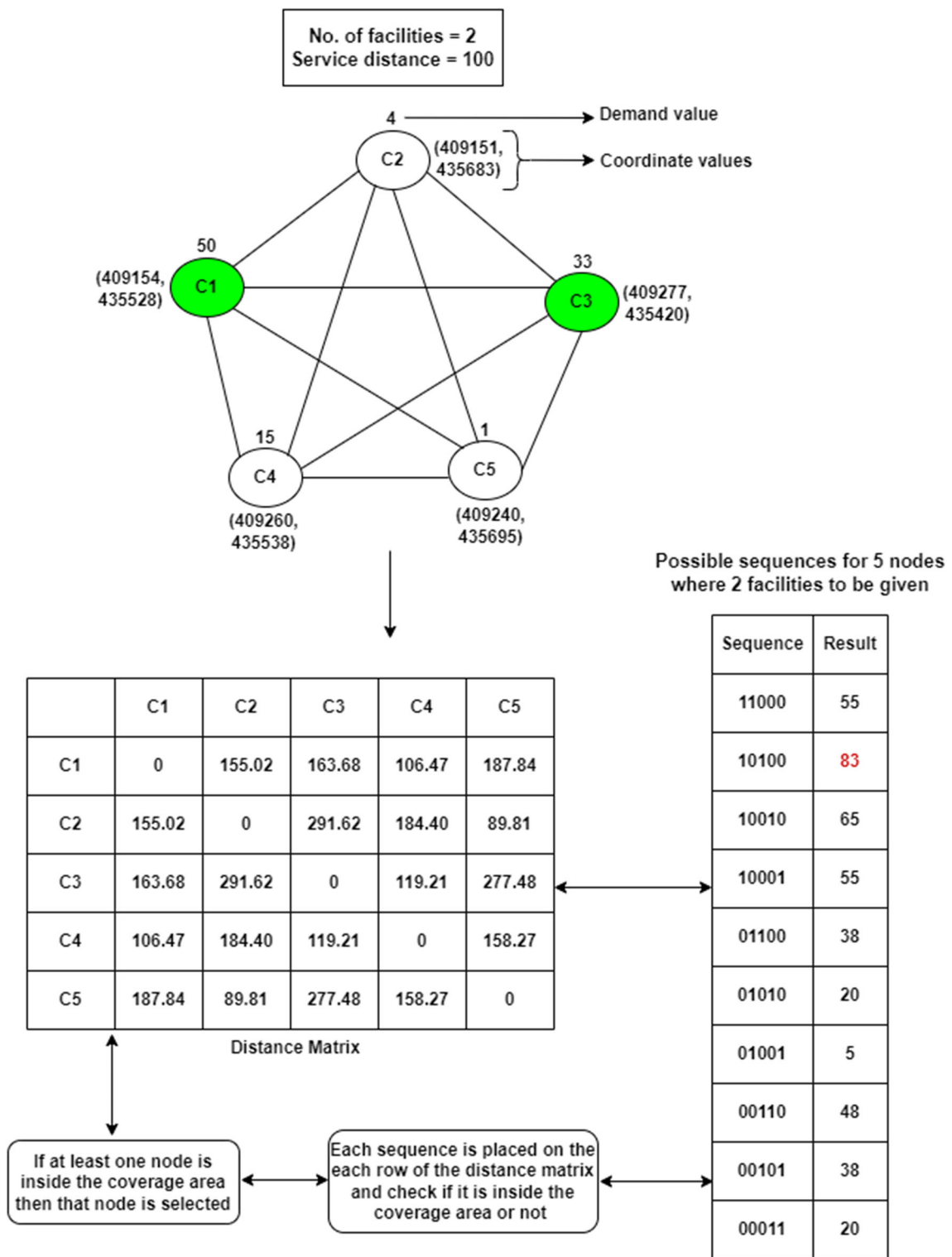


Fig. 1 A sample example of MCLP

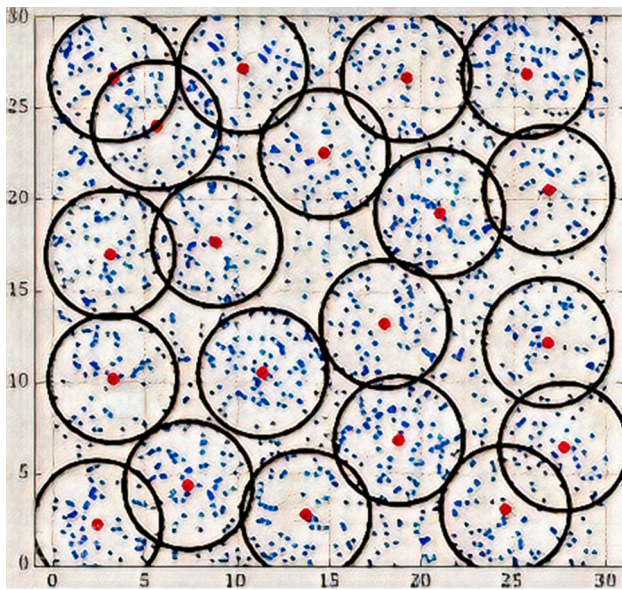


Fig. 2 A sample solution to MCLP

of MCLP. It is tested with 100 to 2500 node networks, and it can give satisfactory results in almost all test instances. The computational cost is relatively high for some instances. This algorithm is a state-of-the-art method to solve the MCLP.

3.2 Greedy randomized adaptive search procedure

Maximo VR et al. proposed an intelligent-guided adaptive search technique to solve the maximum covering location problem (MCLP) in 2016 Bagherinejad et al. (2018). The authors themselves named their proposed method as intelligent-guided adaptive search (IGAS), and it works according to process of the greedy randomized search procedure (GRASP). They created the construction phase of IGAS using artificial neural network (ANN). This method is specialized for large-sized instances (more than 3000 nodes). They used an unsupervised machine learning technique, GNG (neural gas), to monitor the best solutions developed in the current iteration. In this way, this method can learn about the improvement of solutions that helps in the rest iterations. So, their proposed algorithm tries to take decision in such a way that it can traverse through a promising branch. The proposed IGAS is tested with two different experiments. It is compared with GRASP and CPLEX methods with data instances having at least 2700 nodes.

The principal advantage of this strategy is the creation of IGAS which is a variant of GRASP. This method incorporates an adaptive memory so that it can learn from previous solutions. IGAS gives better results almost in all the instances in both experiments. It is a very complex algorithm as hybridization process is applied. The iteration independence of the GRSSP has been violated here.

3.3 Simulated annealing

Zarandi MHF et al. proposed an approach to solve the large-scale dynamic maximal covering location problem in 2013 Zarandi et al. (2013). This paper proposed a long-term version of MCLP named as dynamic MCLP (DMCLP). They used simulated annealing (SA) algorithm, a well-known metaheuristics, to solve the problem. Designing solution representation strategy is the first step of SA. For DMCLP, they used a binary vector representation where each bit denotes the status of a node. In a solution for this representation 1 represents a facility is sited to the node and 0 otherwise. The authors used roulette wheel selection (RWS) technique to accomplish the initialization process of their proposed method. The initial temperature, an important parameter of SA, was set using the method of Crama and Schyns (2003). Three types of cooling schedules (linear cooling, exponential cooling and hyperbolic cooling) were used for DMCLP. They used neighborhood search structure (NSS) to search for better solutions. Their proposed algorithm is tested with random datasets of 1800, 2000 and 2500 nodes along with 100 facility sites. This metaheuristic is compared with CPLEX to measure the accuracy and computational time.

The great achievement of this method is its runtime. It takes less computational time than the existing algorithms to solve the large scale MCLPs. It cannot produce optimal solutions in most of the instances though the gap between the produced solution and the optimal solution is very small.

3.4 Fuzzy simulation

In 2011, Davari S et al. proposed a technique to solve the maximal covering location problem (MCLP) with fuzzy travel times (Davari et al. 2011). They solved the problem by using a hybrid intelligent algorithm where they combined both fuzzy simulation (FS) and simulated annealing (SA) techniques. The authors named their proposed intelligent method as fuzzy version of maximal covering location problem (FMCLP). The travel time between the nodes is referred to as fuzzy variables. SA is a local search-based metaheuristic that searches the solution space stochastically. At the beginning of FMCLP, the SA part generates the initial population with several solutions. FMCLP solution representation is done using binary array representation between the candidate nodes. The travel time matrix is essential along with other general information (the coordinate values and the demand value of each node) of MCLP for this method. The method of Crama and Schyns was used to set the initial temperature of the method (Crama and Schyns 2003). NSS (neighborhood search structure) is used to search for better solutions. The proposed algorithm is tested with random datasets of 50, 100, 200, 500 and 900 nodes, respectively. The test results were compared with those of with an exact method LINGO.

Table 3 Summary of previous works

Name of work	Dataset used	Method	Evaluation	Advantage	Disadvantage
Atta et al. (2018)	SJC324; SJC402 SJC500; SJC708 SJC818; pmed32 pmed39; G&R100* G&R150*; ZDS1800* ZDS2500*	GA	Max Avg Max time Avg time	1. It is almost an optimal and efficient approach that can solve both small and large scale versions of MCLP 2. It is the state-of-art method for MCLP	1. The computational cost is relatively high for some of test instances
Máximo et al. (2017)	vma400; vma600 vma800; vma3038	GRASP IGAS	Best Worst Median Avg time	1.The main advantage of this strategy is the creation of IGAS which is a variant of GRASP 2. It incorporates an adaptive memory	1. The iteration independence of the GRASP has been violated
Zarandi et al. (2013)	mss1800*; mss2000* mss2500*	SA	Best Avg time	1. The great achievement of this method is its runtime 2. It introduces large scale dynamic version of MCLP	1. It cannot produce optimal solutions in most of the instances
Davari et al. (2011)	sma50*; sma100* sma200*; sma500* sma900*	SA FS	Best Avg Worst Avg time	1. It includes the fuzzy theory to MCLP. Other metaheuristics have a chance to bring out a revolution to the fuzzy version of MCLP	1. It cannot produce better results in several test instances
Zarandi et al. (2011)	ZDS1800*; ZDS2500*	GA	Best Avg time	1. This method took less computational time compared to CPLEX method	1. It cannot produce optimal results in several test instances

The proposed method includes the fuzzy theory to MCLP. Other metaheuristics have a chance to bring out a revolution to the fuzzy version of MCLP. The proposed method cannot produce better results in several instances compared to the exact method LINGO with respect to both optimal solution and computational time.

3.5 Genetic algorithm

Zarandi MHF et al. proposed an approach to solve the large-scale maximal covering location problem (MCLP) in 2011 Zarandi et al. (2011). They tried to solve the MCLP for larger number of nodes that's why they named the method as large-scale MCLP. Their proposed method is applicable for up to 2500 nodes. They used the genetic algorithm (GA) to solve the problem. This method applied a binary vector represen-

tation to complete the chromosome encoding. The authors redesigned the selection, crossover and mutation operators of GA They used roulette wheel selection (RWS) technique to design the selection operator. Two random datasets were used to test the results of the proposed algorithm of 1800 and 2500 nodes, respectively. The test results were compared with CPLEX method to observe the performance of the method.

This method took less computational time compared to CPLEX method; it is the achievement of their proposed method. This algorithm could not produce optimal solutions in several instances though the gap between the produced solution and the optimal solution is relatively small.

Table 4 Summary of used datasets in previous works

Name of work	Dataset Used	No. of nodes	Values of facilities	Values of service distance
Atta et al. (2018)	SJC324	324	[1, 2, 3, 4]	[800, 1200, 1600]
	SJC402	402	[1, 2, 3, 4, 5, 6]	[800, 1200, 1600]
	SJC500	500	[1, 2, 3, 4, 5, 6, 7, 8]	[800, 1200, 1600]
	SJC708	708	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]	[800, 1200, 1600]
	SJC818	818	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]	[800, 1200, 1600]
	pmed32	700	[20, 24, 28]	[13, 15, 20]
	pmed39	900	[20, 24, 28]	[10, 13, 16]
	G&R100	100	[8, 10, 12]	[50, 65, 80]
	G&R150	150	[8, 10, 12] and	[75, 80, 85, 90]
			[5, 6, 7, 8, 10, 12, 14, 16, 18, 20]	
Máximo et al. (2017)	ZDS1800	1800	[15, 20, 25]	[3.5, 3.75, 4]
	ZDS2500	2500	[15, 20, 25]	[3.5, 3.75, 4]
	vma400	400	[50, 60, 70, 80, 90, 100]	N/A
	vma600	600	[50, 60, 70, 80, 90, 100]	N/A
Zarandi et al. (2013)	vma800	800	[50, 60, 70, 80, 90, 100]	N/A
	vma3038	3038	[50, 60, 70, 80, 90, 100]	N/A
	mss1800	1800	[65, 70, 75, 80, 85]	[4, 4.5, 5]
	mss2000	2000	[65, 70, 75, 80, 85]	[4, 4.5, 5]
Davari et al. (2011)	mss2500	2500	[65, 70, 75, 80, 85]	[4, 4.5, 5]
	sma50	50	[1, 2]	N/A
	sma100	100	[2, 5]	N/A
	sma200	200	[3, 8]	N/A
	sma500	500	[10, 15]	N/A
Zarandi et al. (2011)	sma900	900	[10, 15]	N/A
	ZDS1800	1800	[15, 20, 25]	[3.5, 3.75, 4]
	ZDS2500	2500	[15, 20, 25]	[3.5, 3.75, 4]

3.6 Summary of related works

We have shown the summary of the literature being studied in Table 3 where we have summarized the previous developed techniques, type of dataset used, evaluation measures, advantages and disadvantages of that technique. The * sign is used after some datasets to represent it as a randomly created dataset. It is evident from Table 3 that most of the previous methods used some random datasets for evaluating their developed methods. The brief summary of the used datasets in previous works is shown in Table 4.

4 Chemical reaction optimization for MCLP

The collective and general term optimization was basically used in lots of scientific fields and applications to come up with best values in different contexts. The goal of optimization can be minimization, maximization, cost reduction, performance upgradation etc. It depends on the problem

definition, objective function and many other catalysts as well (Al-Janabi and Alkaim 2022). The maximal covering location problem is also an optimization problem where optimization means maximization of the total demands of the served population. This section includes several subsections. They are chemical reaction optimization, basic structure of proposed method, solution and population generation, operator design and flowchart of the proposed method.

4.1 Chemical reaction optimization

Lam AYS et al. proposed an algorithm for optimization problem named chemical reaction optimization (CRO), which is a metaheuristic based on nature (Lam and Li 2012). The working principle of CRO actually follows the two principles of thermodynamics. The gist of the first principle (law of conservation) is that energy cannot be produced or spoiled, energy can only be converted from one form to another. So, the amount of total energy remains constant always. This can

be shown by the equation as follows:

$$Popsize(tm) \sum_{x=1} (PE_x(tm) + KE_x(tm) + buffer(tm)) = S \quad (6)$$

In equation (6), $PE_x(tm)$ and $KE_x(tm)$ demonstrate the potential energy (PE) and kinetic energy (KE) of molecule x , respectively, at any time tm . $Popsize(tm)$ is the number of all molecules; $buffer(tm)$ is the energy in the central buffer at time tm . The amount of total constant energy is denoted by S . The value of S validates that CRO follows the energy conservation rule. The second principle of thermodynamics ensures transformation of energy among the molecules (PE is converted to KE during iteration stages). CRO follows these two rules to come up with a better solution for optimization problems.

CRO is a metaheuristic based on population. Basically, CRO performs three core stages. These are initial, iteration and final stages. In the initialization stage, CRO initializes several attributes (initial parameters) and creates the initial population. Each molecule has several attributes. Some of these attributes are molecular structure (α), potential energy (PE), kinetic energy (KE), number of hits ($NumHit$), etc. After initializing the initial parameters, CRO creates the initial population with popsize. Second stage is the iteration stage. Here, one reaction operator (out of four operators) works in each iteration by checking certain conditions. CRO has four elementary reaction operators. They are on wall ineffective collision, intermolecular ineffective collision, synthesis and decomposition. There are no hard and fast rules for designing these operators. The on-wall ineffective collision and the intermolecular ineffective collision ensure intensification (local search). On the contrary, the synthesis and decomposition operators ensure diversification (global search). These four reaction operators make CRO a better method for optimization problems compared to other metaheuristics because of its capability of searching. In addition, CRO allows additional repair operators to be included in the algorithm if needed. Another interesting thing about CRO is that the number of total molecules is not same always during iterations. CRO increases or decreases the number of total molecules according to the current need of the problem. The third one is the final stage of CRO. If any of the stopping criteria is matched during iterations or the limit of iteration exceeds, then CRO goes to the final stage and shows the necessary outputs.

4.2 Basic structure of proposed method

In this article, the MCLP has been solved using chemical reaction optimization algorithm, and hence, the proposed method is named as MCLP_CRO. The initial parame-

Algorithm 1: Init-Pop

```

1 for  $i = 0$  to  $Popsize$  do
2   counter  $\leftarrow$  0
3   while counter  $\leq$   $f - 1$  do
4     rand_num  $\leftarrow$  rand() %  $m$ 
5      $x =$  rand_num
6     if ( $molecules\_vector[i].d[x] == 0$ ) then
7        $molecules\_vector[i].d[x] = 1$ 
8       counter += 1
9     end
10  end
11   $molecules\_vector[i].index = i$ 
12   $molecules\_vector[i].KE = Initial\_KE$ 
13   $molecules\_vector[i].NumHit = 0$ 
14   $molecules\_vector[i].MinHit = 0$ 
15 end

```

ters (Table 5) of CRO are initialized with proper values and then create the initial population (Algorithm 1) using a random function. The pseudo-code of the proposed MCLP_CRO method is shown in Algorithm 2. By passing the initial parameters and initial population to the function MCLP_CRO, the iterations are performed. After the termination of algorithm MCLP_CRO, it measures the outputs. The outputs are saved into two variables named *result* and *time*. The *result* demonstrates the percentage of coverage value, and *time* describes the computational time of the proposed method.

4.3 Solution and population generation

CRO is a metaheuristic method that works on population. A unit from the entire population is called a molecule. The molecule is also known as the manipulated agent due to its actual manipulation to retrieve solution. All the molecules with popsize form the entire population. Every molecule has several parameters. Some of the initial parameters with their corresponding algorithmic definitions and initial values are shown in Table 5. How the initial parameters are tuned to their initial values; the answer of this question is briefly given in Sect. 5.1.

The initial population is created randomly. Algorithm 1 shows the generation process of initial population for the proposed method. Here, *molecules_vector* is the set of all molecules, *counter* is the variable to count the given facilities to a node, rand() is the function that generates a random integer, x is the index of initial potential facility sites, $d[1,2,3,\dots,m]$ is the initial satisfied demand vector of each customer. There are m customers where f facilities to be installed. Each molecule of the initial population is created by locating f random indices from the set of customers or nodes $\{1, 2, 3, \dots, m\}$. Here the value of f is less than m . Each f is generated randomly (random integer mod m). By looping through 0 to *popsize*, all the molecules are

Table 5 Parameters of CRO for MCLP with their initial values

Symbol	Algorithmic definition	Value
PopSize	Size of population	10
KELossRate	Loss rate of kinetic energy	0.2
MoleColl	Collision rate of molecule	0.4
Buffer	Initial energy in the surroundings	0
InitialKE	Initial kinetic energy	1000
alpha (α)	Threshold value for unimolecular reaction	5
beta (β)	Threshold value for bimolecular reaction	15000
NumHit	Number of total hits	0
Iteration	Total number of times the reaction occurs	150
MinPE	The potential energy with minstruct	0
MinHit	The number of hits with minstruct	0

Algorithm 2: MCLP_CRO

```

1 Set initial parameters of CRO: Popsiz, KELossRate, MoleColl,
  buffer, initialKE,  $\alpha$ ,  $\beta$ 
2 Set initial parameters of MCLP: m, r, D[1, 2, ...,m], f
3 Randomly create the initial population with Popsiz
4 Sort the initial population in descending order according to the PE
5 while stopping criteria not met do
6   Generate  $b \in [0, 1]$ 
7   if ( $b > MoleColl$ ) then
8     Randomly Select One Molecule  $r$ 
9     if ( $NumHit_r - MinHit_r > \alpha$ ) then
10      | Trigger Decomposition ( $r$ )
11    end
12    else
13      | Trigger On Wall Ineffective Collision ( $r$ )
14    end
15  end
16  else
17    Randomly Select Two Molecules  $r_1$  and  $r_2$ 
18    if ( $KE_{r1} \leq \beta$  and  $KE_{r2} \leq \beta$ ) then
19      | Trigger Synthesis ( $r_1, r_2$ )
20    end
21    else
22      | Trigger Inter-molecular Collision ( $r_1, r_2$ )
23    end
24  end
25 end
26 for  $r$  in molecules_vector do
27   | Trigger Repair Operator LM-GM ( $r$ )
28 end
29 Find the molecule ( $r$ ) with largest PE
30 Overall best solution saved in result and time variables

```

created accordingly. Thus, the initial population is created. The pseudo-code of our proposed MCLP_CRO approach is depicted in Algorithm 2 where several parameters are used, all the initial parameters of CRO and MCLP are described briefly in Tables 2 and 5, respectively. Here, r represents a molecule; b is a random fractional number in the range [0, 1]. In the proposed MCLP_CRO algorithm, we have used Decomposition(), On Wall Ineffective Collision(), Synthe-

Table 6 Indexed facility (f) array

Index	1	2	3
Value	2	6	8

Table 7 Sample solution representation

Index	1	2	3	4	5	6	7	8	9	10
Value	0	1	0	0	0	1	0	1	0	0

sis(), Inter-molecular Collision() functions as basic CRO operators and LM-GM() as repair operator. In every iteration, either unimolecular or bimolecular reaction will activate depending on the value of threshold parameter *MoleColl* and random fractional parameter b . One of the reaction function operates in each iteration and tries to come up with a good solution. After the iterations stage, LM-GM() is performed to all the molecules to remove the trapping situation of some molecules. Section 4.4 briefly describes the working principle of the reaction functions.

4.3.1 Solution representation

A solution for MCLP is a set of f potential locations those need to be selected from the set of m customers. Let the values of m and f are 10 and 3, respectively. Here, m and f both are represented by one-dimensional arrays. Initially all the values of these two arrays are zero. Let the randomly selected indices for the potential facility sites are 2, 6 and 8. The indexed facility f array for this sample example is shown in Table 6.

A binary vector representation method is used for solution representation. In the solution, the selected indices of f array (Table 6) are denoted by 1, and rest of indices are denoted by 0. Finally, the sample solution representation is shown in Table 7.

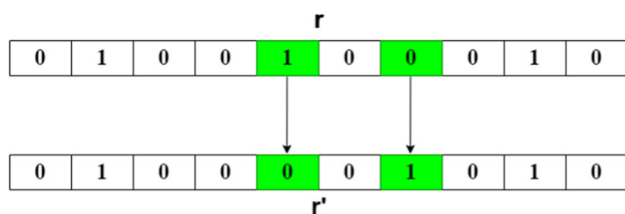


Fig. 3 On-wall ineffective collision

4.4 Operator design

We redesigned the four fundamental operators and an additional repair operator for retrieving the solutions with better results. The operators used in the proposed method are described in the following subsections.

4.4.1 On-wall ineffective collision

If $(NumHit_r - MinHit_r) > \alpha$ satisfies (Algorithm 2), then this reaction operator is operated. When one molecule collides with the wall of a container, then the internal structure of the molecule changes. Here molecule r produces a new molecule r' :

$$r \rightarrow r'$$

Let the values of m and f are 10 and 3. So, each customer needs to provide 3 facilities. The mechanism for on-wall ineffective collision is very simple.

Randomly select one or more indices from molecule r and change it to create a new molecule r' if it ensures better fitness value. In Fig. 3, we can see the 5th and 7th indices of molecule r are changed to form new molecule r' as the new molecule is ensuring a solution with better objective function value.

4.4.2 Decomposition

If $(NumHit_r - MinHit_r) > \alpha$ satisfies (Algorithm 2), then this reaction operator is operated. In this elementary reaction, two new molecules are created from one molecule. Two newly generated molecules bring diversity in their structures from the old molecule. Let molecule r produces two new molecules r_1 and r_2 :

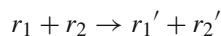
$$r \rightarrow r_1 + r_2$$

According to our example, we have 10 customers and each customer needs to be provided with 3 facilities. The mechanism is quite simple. Firstly, divide the molecule r into two partitions using a random divider function. Then, the first and second partition of molecule r are copied to the beginning indices of molecules r_1 and r_2 , respectively. The rest of the

indices of molecules r_1 and r_2 are created randomly using a random function generator while maintaining the constraints of the problem definition. In Fig. 4, the initial three indices of molecule r are copied to form a new molecule r_1 and rest of the seven indices of molecule r are copied to the initial indices of molecule r_2 . The 4th to 10th indices of molecule r_1 are created randomly while maintaining the constraints of the problem definition properly. Same mechanism is carried out for molecule r_2 as well.

4.4.3 Intermolecular ineffective collision

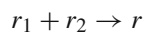
If $(KE_{r_1} > \beta$ and $KE_{r_2} > \beta)$ satisfies (Algorithm 2), then this reaction operator is operated. Here, two molecules collide with each other to form two new molecules. Let two molecules r_1 and r_2 collide with each other and produce two new molecules r_1' and r_2' .



This is much similar to on-wall ineffective collision except that the number of molecules is twice here. Molecule r_1' is produced from molecule r_1 , and molecule r_2' is produced from r_2 . The mechanism is same as on-wall ineffective collision. Several indices are changed to form a new molecule. In Fig. 5, the 6th and 8th indices of molecule r_1 are changed to form a new molecule r_1' and rest of the indices of molecule r_1 are copied to the same indices of molecule r_1' . Same operations are carried out for creating molecule r_2' as well.

4.4.4 Synthesis

When $(KE_{r_1} \leq \beta$ and $KE_{r_2} \leq \beta)$ satisfies (Algorithm 2), then this reaction operator is activated. Synthesis consolidates two molecules to form a new molecule. It is a reverse procedure of decomposition. Let r_1 and r_2 be two molecules. After the collision, molecule r is created.



Synthesis performs diversification to travel the global solution space and tries to retrieve the optimal solution. One point crossover mechanism is used for synthesis. The first few indices of molecule r are copied from molecule r_1 and rest of the indices are copied from molecule r_2 to form the new molecule r . In Fig. 6, the first four indices of molecule r_1 and the first six indices of molecule r_2 are copied to form a new molecule r which carries diversity.

4.4.5 Repair operator LM-GM

In some cases, the MCLP_CRO algorithm trapped into local maximum, so the optimal results cannot be obtained. To over-

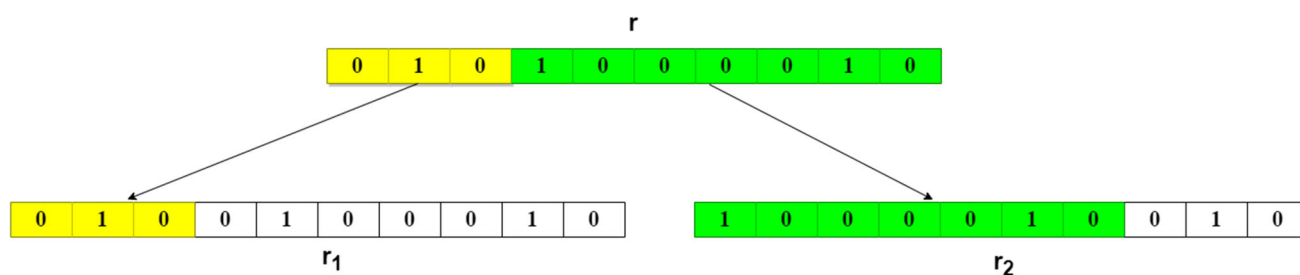


Fig. 4 Decomposition

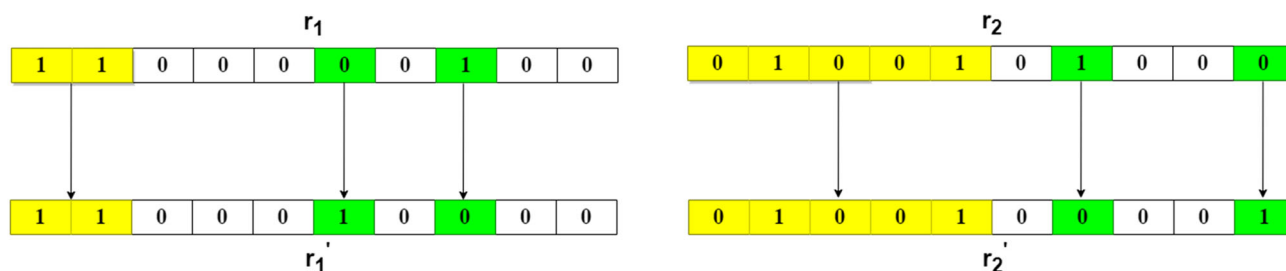


Fig. 5 Intermolecular ineffective collision

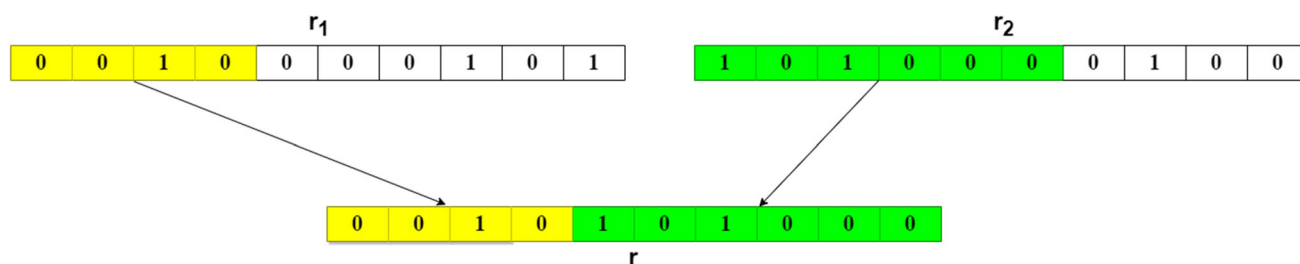


Fig. 6 Synthesis

come the trapping situation, we have designed a new repair operator named LM-GM, which converts local maximum into global maximum. Pseudo-code of LM-GM operator is shown in Algorithm 3. Determining deviation indices (line 10–16 in Algorithm 3) of each molecule is carried out first, and then, the operator performs the demand maximization (line 17–22 in Algorithm 3) with the help of deviation indices. The working principle of LM-GM is depicted in Fig. 7. Sometimes current solution get trapped into local maximum because the basic reaction operators were designed using several random functions. For solving this issue, an iterative max deviation index search is done in LM-GM operator. In Fig. 7, the max deviation indices of 1st and 9th elements of the original solution are mapped to 6th and 10th indices to form a new solution. The 4th index remains unchanged in the new solution because it already holds the max deviation index. New solution produced the max point of MCLP.

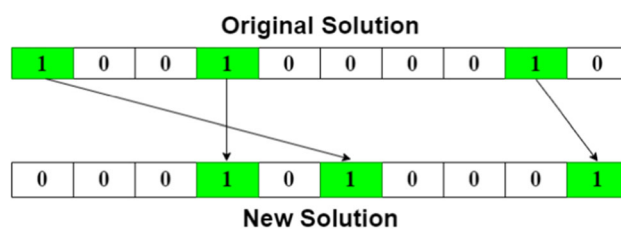


Fig. 7 Repair operator LM-GM

4.5 Flowchart of the proposed method

Figure 8 depicts the flowchart of the proposed method. We have created the initial population using a random function generator, and the pseudo-code of the process is given in Algorithm 1 and the description is provided in Sect. 4.3. Before starting iterations, sort all the molecules of initial population according to their potential energies. Then in iterations stage, check whether the reaction is intermolecular or unimolecular. If intermolecular reaction occurs ($b >$

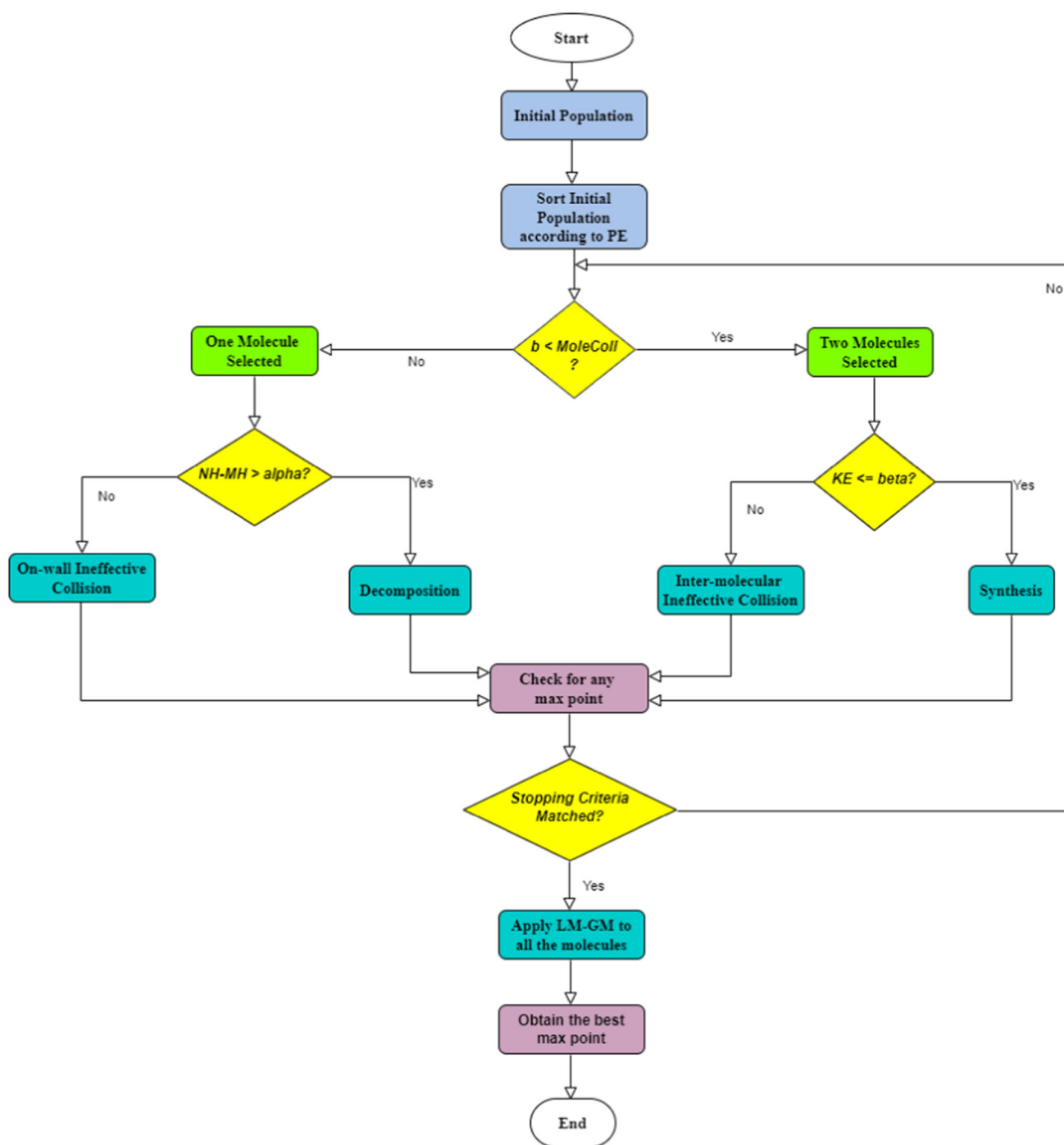


Fig. 8 Flowchart of the proposed MCLP_CRO method

$MoleColl$ satisfies), then either do synthesis (see Sect. 4.4.4) or intermolecular ineffective collision (see Sect. 4.4.3) by checking which one is appropriate. If $KE > beta$ (means current kinetic energy exceeds the threshold beta), then execute the global search strategy synthesis, otherwise go for the local search technique intermolecular ineffective collision to hold the current state of the molecules maximized and balanced. When $b < MoleColl$, perform decomposition (see

Sect. 4.4.2) or on wall ineffective collision (see Sect. 4.4.1) in the same manner. In Fig. 8, NH and MH refer to the number of total hits and the number of hits with minstruct, respectively. When $NH - MH > alpha$, then the algorithm asks for a global search strategy decomposition to activate. Otherwise perform the local search method on wall ineffective collision to develop the molecules locally. After each iteration, check whether a max point is found or not. If any max

Algorithm 3: LM-GM

```

1 Take each molecule x iteratively one by one as parameter
2 Take an initial parameter deviation array
3 Take a molecule type variable y
4 sum ← 0
5 y ← x
6 for i = 0 to m do
7   sum += molecule[i]
8 end
9 while sum <= f do
10  for i = 0 to m do
11    if (y.molecule[i] == 0) then
12      y.molecule[i] = 1
13      deviation[i] = y.PE - current.PE
14    end
15    y = x
16  end
17  for i = 0 to m do
18    if (deviation[i] > Max_demand) then
19      Max_demand = deviation[i]
20      index = i
21    end
22  end
23  x.molecule[index] = 1
24  sum += 1
25 end
26 Outputs: Get the molecules with max deviation index

```

point is found, then verify the stopping criteria, if it matched, then obtain the best max point and terminates. Otherwise, again check from the beginning, that is whether the reaction is intermolecular or unimolecular. Perform the same again until get a max point or reach the iteration limit.

5 Experimental results and comparisons

Now data are one of the world's most valuable resources that creates several fields of computer science. Without dataset, any expert or digital system cannot be established. Data can be of various types and can be gathered by recording, search or observation. We have taken nine datasets (five real-world datasets and four randomly created datasets) to measure the performance of proposed (MCLP_CRO) algorithm. The five real-world datasets are named as SJC_324, SJC_402, SJC_500, SJC_708 and SJC_818 where 324, 402, 500, 708 and 818 are the total number of nodes in the datasets, respectively. On the other hand, four random datasets are named as B_700, B_900, ZDS_1800, and ZDS_2500 where 700, 900, 1800, and 2500 are the number of nodes in the datasets. These datasets were used for evaluating recent algorithms for solving maximal covering location problem (Atta et al. 2018; Lorena and Pereira 2002). We implemented our proposed algorithm (MCLP_CRO) by using C++ language with device specifications: Processor- Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz, RAM - 8.00 GB (7.90 GB usable),

Table 8 Selected instances for tuning parameter

n	p	s	Instance name	Instance size
324	1	800	N324_1F	Small
324	5	800	N324_5F	Small
402	6	800	N402_6F	Medium
500	8	800	N500_8F	Medium
708	11	800	N708_11F	Large
818	14	800	N818_14F	Large

System type - 64-bit operating system, x64-based processor, OS - Windows 10 Pro edition.

5.1 Parameter tuning

It is important to tune the initial attributes of CRO to obtain the better results. We tuned several necessary attributes of CRO such as α , β , kinetic energy (KE), and the number of runs. Six instances were taken for the tuning: two large, two medium and two small instances, which are shown in Table 8.

For α tuning, we have taken the values of α from 1 to 8. In Fig. 9, the values of α are denoted along X axis and the values of potential energy of each instance are shown along Y axis. Figure 9 illustrates the value of tuning for α . When the value of α is 5, we got the highest potential energy in most cases. In addition, the highest potential energies for the selected six instances (Table 8) are 5461, 12152, 15984, 19707, 24192 and 29168 found (Fig. 9) when α is 5. Therefore, we used 5 as the initial value of α for the proposed MCLP_CRO method.

The values of β have been taken from 12000 to 18000 for β tuning. In Fig. 10, the values of β are represented along X axis and the values of potential energy of each instance are depicted along Y axis. Figure 10 demonstrates the value of tuning for β . It can be noticed that when the value of β is 15000, we got highest potential energy in most cases. Rather, the highest potential energy for the selected six instances (Table 8) are 5461, 12152, 15984, 19707, 24192 and 29168 found (Fig. 10) when β is 15000. So, we used 15000 as the initial value of β for the proposed MCLP_CRO algorithm.

For the tuning purpose of kinetic energy (KE), we have considered the values of KE from 400 to 1600. In Fig. 11, the values of KE are represented along X axis and the values of potential energy of each instance are shown along Y axis. Figure 11 indicates the value of tuning for KE. It is noticed that when the value of KE is 1000, we obtained highest potential energy in most cases. In addition, the highest potential energy for the selected six instances (Table 8) are 5461, 12152, 15984, 19707, 24192 and 29168 found (Fig. 11) when KE is 1000. So, we selected 1000 as the initial value of KE for the proposed MCLP_CRO approach.

Fig. 9 Alpha tuning

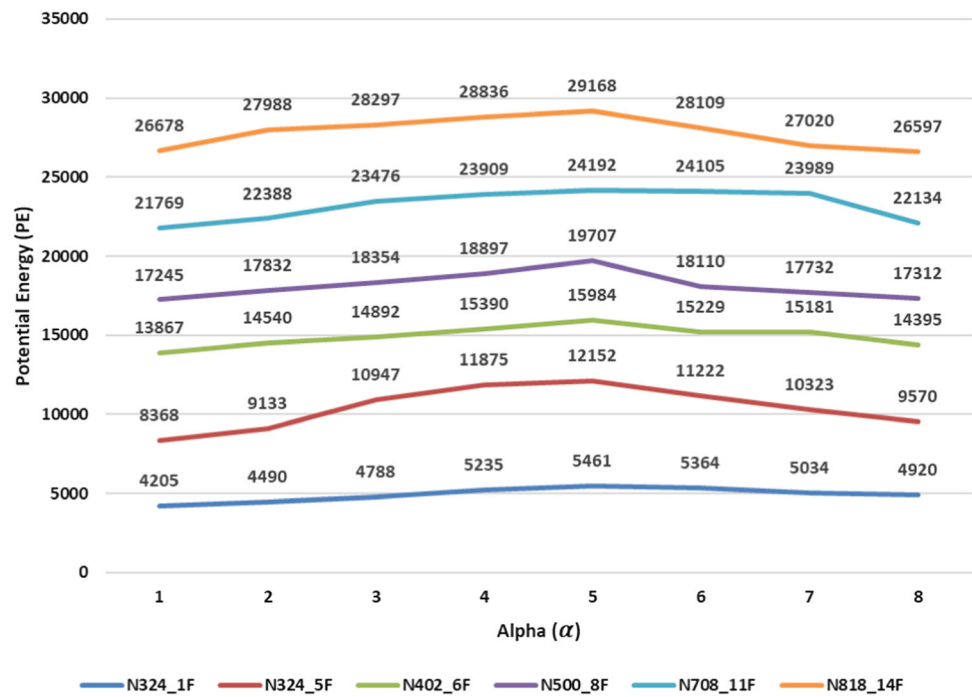
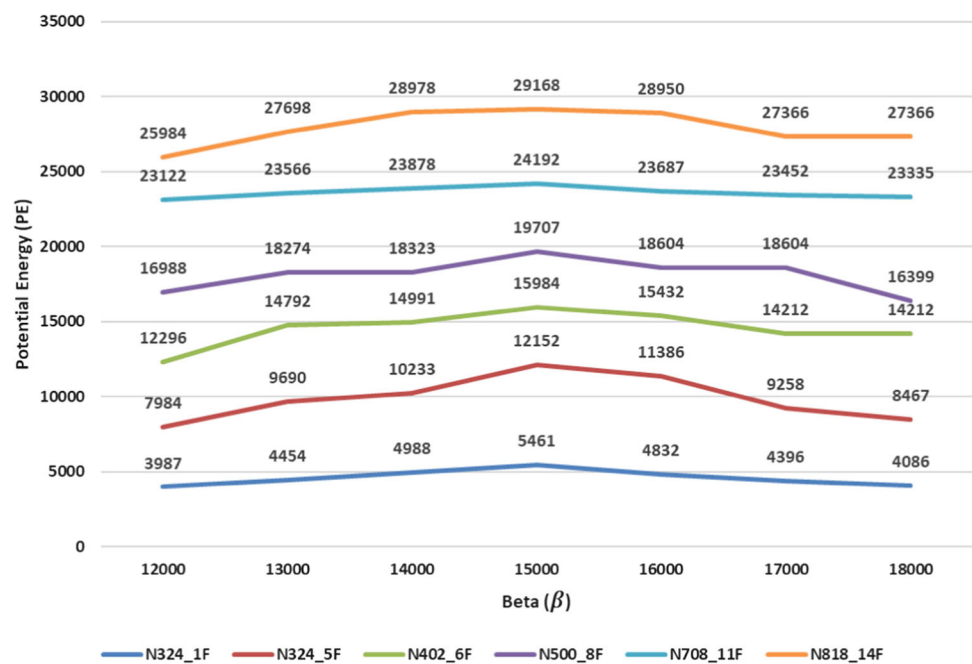


Fig. 10 Beta tuning



To observe the significance of the number of runs, we have examined the instances 5, 10, 15, 20, 25, and 30 times. When the number of runs is over 15, the potential energy remains same in most cases as no new value found. In Fig. 12, the values of number of runs are represented along X axis and the values of potential energy of each instance are depicted along Y axis. Figure 12 indicates the effect of number of runs over the potential energy.

5.2 Results analysis and discussion

We initialized the CRO attributes as $PopSize = 10$, $KELossRate = 0.2$, $MoleColl = 0.4$, $InitialKE = 1000$, $\alpha = 5$ and $\beta = 15000$, $iteration = 150$. The input parameters are recorded as n , p and S , where n denotes the number of customers or nodes in the network, p is the number of facilities to be installed and S represents the constant service distance or service radius within the facilities that can be provided to a customer. These three symbols (n , p and S)

Fig. 11 KE tuning

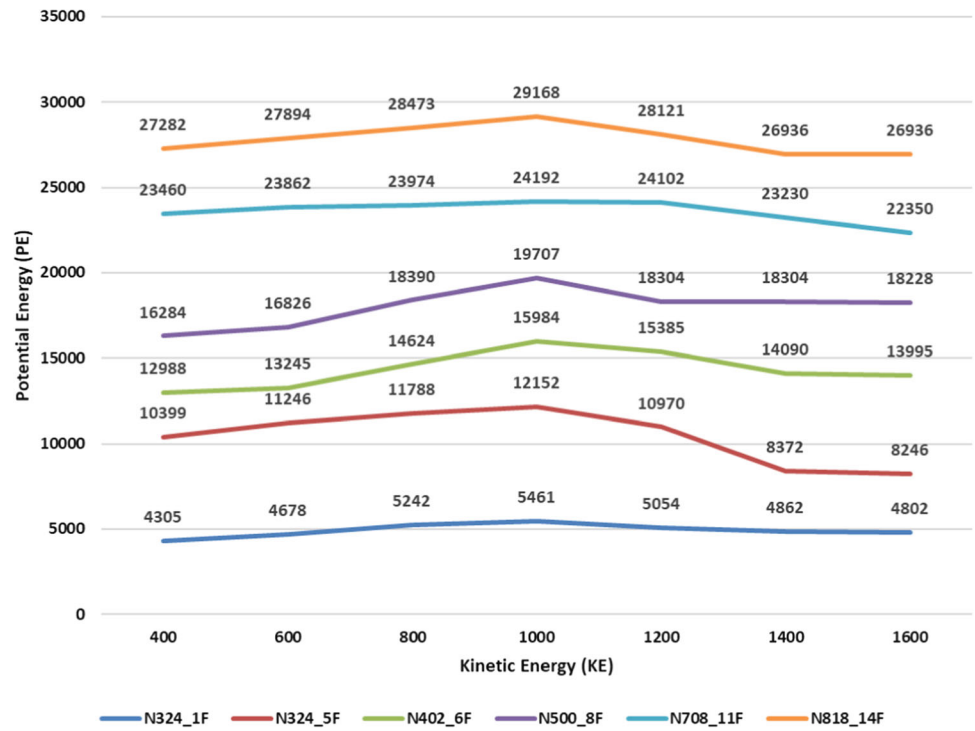


Fig. 12 Tuning of runs

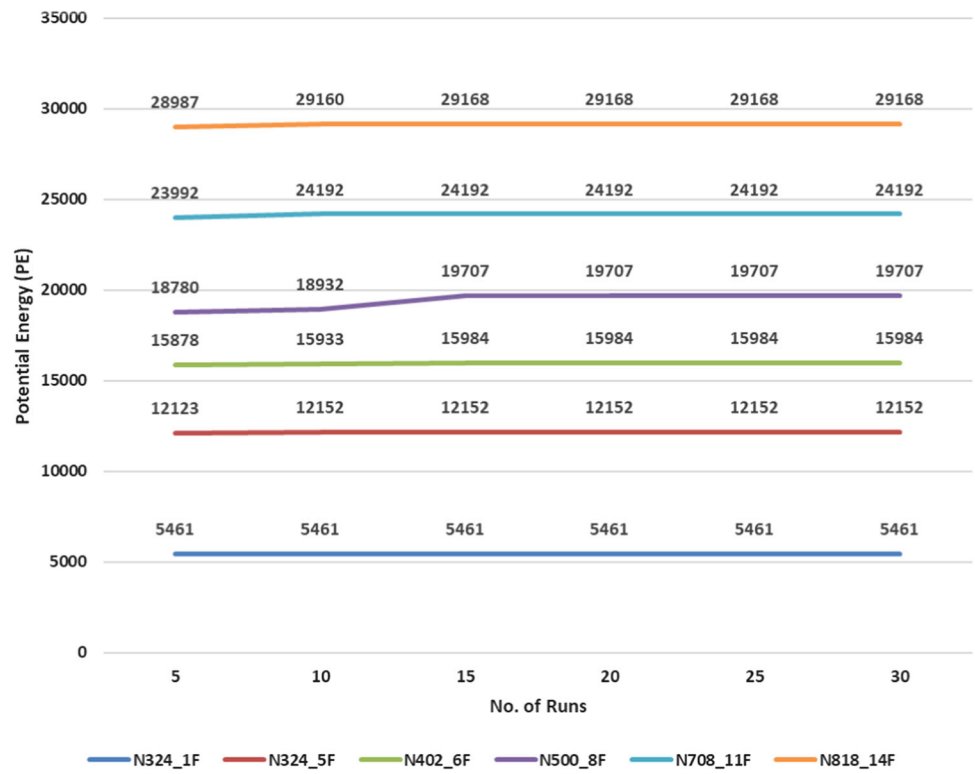


Table 9 Results of SJC_324 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018)

n	p	s	Known best % of cov (Lorena and Pereira 2002)	Atta_GA		MCLP_CRO without LM-GM		MCLP_CRO with LM-GM	
				% of cov	Time (s)	% of cov	Time (s)	% of cov	Time (s)
324	1	800	44.94	44.94	2.42	44.94	1.45	44.94	1.38
324	2	800	72.33	72.33	3.79	72.33	1.41	72.33	1.47
324	3	800	95.49	95.49	5.49	95.49	1.73	95.49	1.32
324	4	800	99.62	99.62	5.07	99.62	1.65	99.62	1.34
324	5	800	100	100	6.45	99.80	1.16	100	1.20
324	1	1200	81.73	81.73	2.95	81.73	1.47	81.73	1.64
324	2	1200	95.08	95.08	4.25	95.08	1.08	95.08	1.08
324	3	1200	100	100	5.34	100	1.16	100	1.16
324	1	1600	99.76	99.76	3.51	99.76	1.46	99.76	1.37
324	2	1600	100	100	3.94	100	1.27	100	1.21

Table 10 Results of SJC_402 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018)

n	p	s	Known best % of cov (Lorena and Pereira 2002)	Atta_GA		MCLP_CRO without LM-GM		MCLP_CRO with LM-GM	
				% of cov	Time (s)	% of cov	Time (s)	% of cov	Time (s)
402	1	800	41.01	41.01	4.03	41.01	1.47	41.01	1.41
402	2	800	70.94	70.94	6.40	70.94	1.34	70.94	1.39
402	3	800	91.90	91.90	8.79	91.90	1.33	91.90	1.38
402	4	800	97.85	97.85	8.34	97.96	1.38	97.96	1.29
402	5	800	99.91	99.91	10.52	99.28	1.55	99.91	1.69
402	6	800	100	100	13.36	100	1.73	100	1.71
402	1	1200	66.36	66.36	5.18	66.36	1.48	66.36	1.43
402	2	1200	92.79	92.79	8.86	92.79	1.39	92.79	1.38
402	3	1200	100	100	8.75	100	1.64	100	1.36
402	1	1600	99.58	99.58	5.93	99.58	1.51	99.58	1.48
402	2	1600	100	100	6.92	100	1.66	100	1.69

are used in Tables 8 to 17. The *% of cov* and *Time(s)* represent the percentage of coverage and the computational time in second, respectively. These two symbols (*% of cov* and *Time(s)*) are used in Tables 9 to 13. For a fair comparison, we implemented Atta_GA (Atta et al. 2018) also using C++ in the same machine with the same device specification as for MCLP_CRO. The proposed algorithm is compared with Atta_GA only because this algorithm achieved best results in all the instances, and it is the state-of-the-art method. The bold sign used in Tables 9, 10, 11, 12, 13, 15, 16, 17 denote best achieved results of the proposed method.

The five real-world datasets were collected from "L.A.N.Lorena-instancias - INPE" whose link is available in the website (<http://www.lac.inpe.br/~lorena/instancias.html>). These five datasets were tested with three different values of S . These values are 800, 1200 and 1600. There are 10 instances in SJC_324 dataset. By increasing the value of facility by one unit every time, we have measured the per-

centage of coverage and the computational time. The good results of proposed MCLP_CRO method are highlighted by bold sign. We obtained the same results as Atta_GA (Atta et al. 2018) with respect to percentage of coverage (Table 9). However, the computational time for each of 10 instances is less than that of Atta_GA. Table 10 shows the comparison for SJC_402 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018). We got the same results as Atta_GA (Atta et al. 2018) with respect to percentage of coverage (Table 10). However, the computational time of each of the 11 instances is less than that of Atta_GA.

The results for SJC_500 dataset are shown in Table 11. SJC_500 is a dataset of 500 customers containing 15 instances. For 13 instances out of 15, we have obtained the same results with respect to percentage of coverage compared to Atta_GA (Atta et al. 2018) in less computational time. For 2 instances ($[n = 500, p = 4, s = 800]$ and $[n = 500, p = 5, s = 800]$) we got worse but very near results with respect

Table 11 Results of SJC_500 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018)

n	p	s	Known Best	Atta_GA		MCLP_CRO without LM-GM		MCLP_CRO with LM-GM	
			% of cov (Lorena and Pereira 2002)	% of cov	Time (s)	% of cov	Time (s)	% of cov	Time (s)
500	1	800	40.31	40.31	4.12	40.31	3.21	40.31	2.76
500	2	800	63.20	63.20	4.85	63.20	3.25	63.20	3.12
500	3	800	79.82	79.82	5.96	79.82	3.65	79.82	3.49
500	4	800	90.29	90.29	7.49	89.74	3.45	90.04	3.43
500	5	800	95.70	95.70	9.53	95.02	3.32	95.23	3.36
500	6	800	99.08	99.08	13.26	97.40	3.87	99.08	3.73
500	7	800	99.92	99.92	15.76	99.77	4.73	99.92	4.62
500	8	800	100	100	14.29	100	3.89	100	3.51
500	1	1200	54.43	54.43	4.79	54.43	3.12	54.43	2.39
500	2	1200	91.69	91.69	6.93	91.69	3.24	91.69	2.97
500	3	1200	98.41	98.41	9.37	98.41	3.76	98.41	3.42
500	4	1200	100	100	13.25	100	2.79	100	1.75
500	1	1600	75.12	75.12	5.07	75.12	3.20	75.12	2.89
500	2	1600	99.8	99.8	9.63	99.8	3.56	99.8	3.32
500	3	1600	100	100	12.96	100	3.14	100	2.18

to percentage of coverage. However, the computational time for all the instances is less than that of Atta_GA (Atta et al. 2018).

The comparison results for SJC_708 and SJC_818 datasets between MCLP_CRO and Atta_GA (Atta et al. 2018) are given in Tables 12 and 13, respectively. SJC_708 and SJC_818 are two datasets containing 21 and 26 instances. For 20 instances out of 21 (SJC_708 dataset), we have got same results with respect to percentage of coverage compared to Atta_GA (Atta et al. 2018) in less computational time. For one instance ($n = 708$, $p = 7$, $s = 800$), we got worse but very near results with respect to percentage of coverage. However, the computational time of each of the 21 instances (SJC_708 dataset) is less than that of Atta_GA. Atta et al. (2018). For 22 instances out of 26 (SJC_708 dataset), the same results have obtained with respect to percentage of coverage compared to Atta_GA (Atta et al. 2018), while the computational time of each of the 26 instances is less than that of Atta_GA.

The five real-world datasets were used (SJC_324, SJC_402, SJC_500, SJC_708, SJC_818) by Atta et al. (2018) and Lorena and Pereira (2002) for solving the MCLP. Lorena and Pereira (2002) used Lagrangean/surrogate heuristic for solving MCLP, and to the best our knowledge, this is the first method that used several real-world datasets (GIS-referenced instances of São José dos Campos city of Brazil) to deal with MCLP. Their proposed method performed very well with respect to percentage of coverage for all the test instances. Atta et al. (2018) further used those five real-world datasets to solve MCLP using genetic algorithm with local refine-

ment. They compared their obtained results with the method proposed by Lorena and Pereira (2002) and showed that their method is better than almost all the test instances with respect to both percentage of coverage and computational time. In this paper, we have also used these five real-world datasets, and we have compared our proposed method with the method proposed by Atta et al. (2018). The first five rows of Tables 18 and 19 show the comparison between our proposed method and Atta_GA (Atta et al. 2018) with respect to computational time and percentage of coverage, respectively. The average computational time for SJC_324 dataset is 7.025 s in Lorena and Pereira (2002), 4.32 s in Atta et al. (2018) and our proposed method got 1.32 s which is more efficient than the previous two methods.

We have created four random datasets to measure the performance of the proposed method (MCLP_CRO) for large scales of MCLP. These four datasets were used by Atta_GA (Atta et al. 2018), and these datasets are named as B_700, B_900, ZDS_1800 and ZDS_2500 containing 700, 900, 1800 and 2500 customers (nodes), respectively. For these four datasets, the *Avg Cov %* and *Max Cov %* denote the percentage of average coverage and the percentage of maximum coverage, respectively. The *Avg Time(s)* and *Max Time(s)* denote the average computational time and maximum computational time in seconds. The two datasets pmed32 (B_700) and pmed39 (B_900) were collected from “OR-Library” whose link is available in the website (<http://people.brunel.ac.uk/~mastjib/jeb/info.html>). These two datasets are for pmedian problem, so these datasets do not contain the demand values of the customers. So the demand values of

Table 12 Results of SJC_708 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018)

n	p	s	Known best % of cov (Lorena and Pereira 2002)	Atta_GA		MCLP_CRO without LM-GM		MCLP_CRO with LM-GM	
				% of cov	Time (s)	% of cov	Time (s)	% of cov	Time (s)
708	1	800	34.69	34.69	9.61	34.69	3.07	34.69	3.12
708	2	800	55.00	55.00	12.84	55.00	3.54	55.00	3.51
708	3	800	71.40	71.40	15.21	71.40	4.67	71.40	4.53
708	4	800	84.07	84.07	17.43	83.84	4.23	84.07	4.21
708	5	800	88.81	88.81	19.21	87.44	4.21	88.81	4.28
708	6	800	93.02	93.02	18.26	91.04	5.23	93.02	6.13
708	7	800	95.70	95.70	20.53	94.25	4.88	95.63	5.94
708	8	800	97.83	97.83	23.75	97.33	4.58	97.83	5.39
708	9	800	99.10	99.10	25.36	98.13	4.81	99.10	5.25
708	10	800	99.99	99.99	27.18	98.70	5.15	99.99	5.74
708	11	800	100	100	26.32	99.14	4.23	100	5.87
708	1	1200	48.00	48.00	7.94	48.00	3.13	48.00	3.05
708	2	1200	84.23	84.23	14.96	84.23	4.28	84.23	4.21
708	3	1200	92.68	92.68	15.24	92.68	5.58	92.68	5.37
708	4	1200	98.73	98.73	19.61	96.60	5.45	98.73	5.45
708	5	1200	99.66	99.66	21.08	98.33	4.25	99.66	5.21
708	6	1200	100	100	4.19	23.33	5.03	100	5.16
708	1	1600	69.56	69.56	8.43	69.56	3.82	69.56	3.69
708	2	1600	96.59	96.59	16.39	96.59	4.51	96.59	4.54
708	3	1600	98.59	98.59	21.14	98.74	4.91	98.74	4.27
708	4	1600	100	100	20.26	100	4.32	100	4.17

customers are created using a separate program from a normal distribution with mean 80 and standard deviation 15 as done in Atta_GA (Atta et al. 2018). Tables 14 and 15 show the results for B_700 and B_900 datasets between MCLP_CRO and Atta_GA (Atta et al. 2018). B_700 dataset is tested with 3 different values of S . These values are 13, 15 and 20. On the other hand, B_900 dataset is tested with different three different values (10, 13 and 16) of S . The number of facilities to be installed for both the datasets (B_700 and B_900) are 20, 24 and 28. There are nine instances for these two datasets. For all the instances (B_700 dataset), we have obtained better results with respect to percentage of coverage (both average and maximum) and computational time (both average and maximum) compared to Atta_GA (Atta et al. 2018). For eight instances out of nine (B_900 dataset), we have got better results with respect to average percentage of coverage compared to Atta_GA (Atta et al. 2018). On the contrary, with respect to max percentage of coverage, seven instances out of nine, we have got better results. However, the computational time (both average and maximum) of each of the nine instances is less than that of Atta_GA for B_900 dataset.

The two random datasets were examined (B_700, B_900) by Atta et al. (2018) and Lorena and Pereira (2002) for solving the MCLP. Lorena and Pereira (2002) tested their method

with these two datasets having the demand values of customers those are created randomly using a separate program from a normal distribution with mean 80 and standard deviation 15. Lorena and Pereira (2002) made the first attempt to use these specific types of random datasets to deal with MCLP using Lagrangean/surrogate heuristic. Their proposed method performs well with respect to maximum percentage of coverage on these datasets. Later a metaheuristic based approach were proposed by Atta et al. (2018) which outperform the previous method in almost all the dimensions. In this paper we have also used these two random datasets, and we have compared our proposed method with the method proposed by Atta et al. (2018). The sixth and seventh rows of Tables 18 and 19 show the comparison between our proposed method and Atta_GA (Atta et al. 2018) with respect to average computational time and percentage of coverage, respectively. The average computational time for B_700 dataset is 334.69s in Lorena and Pereira (2002), 43.84s in Atta et al. (2018) and our proposed method obtained 18.59s which is more efficient.

The distance graph of ZDS_1800 and ZDS_2500 datasets are created randomly in the range [0, 30] in uniform distribution. The demand values are also created randomly from a uniform distribution in the range [0, 100]. Two sepa-

Table 13 Results of SJC_818 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018)

n	p	s	Known best % of cov (Lorena and Pereira 2002)	Atta_GA		MCLP_CRO without LM-GM		MCLP_CRO with LM-GM	
				% of cov	Time (s)	% of cov	Time (s)	% of cov	Time (s)
818	1	800	28.77	28.77	8.89	28.77	3.99	28.77	3.97
818	2	800	45.62	45.62	12.26	45.62	5.35	45.62	5.28
818	3	800	60.02	60.02	15.92	60.02	4.12	60.02	4.26
818	4	800	73.46	73.46	16.23	70.52	5.23	73.46	5.25
818	5	800	84.10	84.10	19.21	78.77	6.98	84.10	6.63
818	6	800	88.82	88.82	17.77	84.14	5.18	88.82	5.39
818	7	800	92.34	92.34	23.05	88.86	7.23	92.26	7.11
818	8	800	95.35	95.35	25.71	92.80	5.34	95.35	6.84
818	9	800	97.36	97.36	22.13	95.60	5.26	97.36	7.89
818	10	800	98.55	98.55	26.52	96.76	5.18	98.51	6.76
818	11	800	99.74	99.74	28.16	97.92	5.20	99.74	6.29
818	12	800	99.81	99.81	26.44	98.79	7.16	99.81	7.12
818	13	800	99.98	99.98	24.98	99.13	6.83	99.98	7.89
818	14	800	100	100	33.19	99.38	6.75	100	8.92
818	1	1200	39.81	39.81	7.68	39.81	4.46	39.81	4.29
818	2	1200	69.56	69.56	13.26	69.56	5.53	69.56	5.37
818	3	1200	86.43	86.43	19.88	85.86	4.13	86.39	4.57
818	4	1200	92.67	92.67	16.63	91.48	5.45	92.67	5.87
818	5	1200	97.75	97.75	17.22	95.87	5.75	97.75	5.73
818	6	1200	99.59	99.59	20.49	97.81	5.43	99.59	5.28
818	7	1200	99.96	99.96	24.18	99.23	5.57	99.84	5.65
818	1	1600	57.69	57.69	8.12	57.69	4.34	57.69	4.11
818	2	1600	84.50	84.50	11.35	84.50	4.13	84.50	4.13
818	3	1600	94.87	94.87	12.87	91.34	4.20	94.87	4.27
818	4	1600	98.95	98.95	15.66	95.12	5.56	98.95	5.29
818	5	1600	100	100	18.95	98.49	4.54	100	4.91

Table 14 Results of B_700 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018)

n	p	s	Atta_GA				MCLP_CRO without LM-GM				MCLP_CRO with LM-GM			
			Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)	Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)	Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)
700	20	13	70.63	71.27	37.24	41.51	70.18	71.34	17.19	18.05	71.19	72.50	16.31	17.79
700	24	13	74.25	75.19	42.37	45.13	75.03	75.47	17.53	18.41	75.93	76.95	17.72	18.31
700	28	13	77.69	78.46	48.55	53.66	82.31	82.89	20.39	22.04	82.75	83.44	20.24	21.73
700	20	15	79.8	80.2	39.78	44.74	79.68	80.23	15.32	17.49	80.13	81.38	14.77	16.69
700	24	15	83.07	83.85	46.49	50.11	83.46	83.92	15.69	17.53	84.80	85.81	14.89	16.78
700	28	15	86.25	87.34	44.62	48.94	90.23	90.73	16.17	17.22	91.08	91.69	16.20	17.07
700	20	20	96.28	96.52	36.89	41.93	96.32	96.64	17.17	17.89	96.61	96.99	16.89	17.28
700	24	20	97.23	97.54	48.29	55.76	97.74	98.08	19.31	20.41	98.02	98.35	19.26	20.19
700	28	20	98.12	98.33	50.34	58.75	98.21	98.59	20.71	21.27	99.05	99.14	20.63	21.53

Table 15 Results of B_900 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018)

n	p	s	Atta_GA				MCLP_CRO without LM-GM				MCLP_CRO with LM-GM			
			Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)	Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)	Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)
900	20	10	67.38	67.98	41.99	47.32	67.48	68.02	19.51	20.86	67.52	68.16	19.56	20.17
900	24	10	71.14	71.91	44.78	51.98	71.23	71.97	21.47	22.36	71.34	72.09	21.39	22.14
900	28	10	74.34	75.21	54.25	59.77	74.59	75.15	21.42	22.37	74.86	75.51	21.45	22.46
900	20	13	87.43	87.91	43.22	48.92	87.12	87.58	19.29	19.91	87.75	87.96	19.25	19.87
900	24	13	89.23	90.32	48.57	59.31	88.96	89.82	20.17	20.72	89.32	90.89	19.96	20.59
900	28	13	91.33	92.07	49.44	54.86	91.24	91.86	22.25	23.07	92.06	93.96	22.13	22.97
900	20	16	96.34	96.91	42.97	53.26	96.24	96.89	19.25	19.88	96.42	97.03	19.39	20.65
900	24	16	97.43	97.81	45.99	57.64	97.32	97.63	21.99	22.36	97.46	97.80	21.96	22.27
900	28	16	98.26	98.64	52.41	60.23	98.04	98.51	23.19	23.97	98.19	98.56	23.21	23.93

Table 16 Results of ZDS_1800 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018)

n	p	s	Atta_GA				MCLP_CRO without LM-GM				MCLP_CRO with LM-GM			
			Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)	Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)	Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)
1800	15	3.5	71.01	71.39	47.77	54.98	72.46	72.89	20.17	20.37	73.04	73.13	20.19	21.32
1800	15	3.75	79.01	79.23	48.53	54.22	78.91	79.32	22.39	25.86	79.04	79.96	22.34	25.29
1800	15	4	85.42	85.81	54.97	59.42	86.23	86.89	23.13	25.12	86.90	87.72	23.12	24.36
1800	20	3.5	87.01	87.65	54.89	62.54	87.12	87.58	31.32	31.67	87.74	88.24	30.21	31.54
1800	20	3.75	92.32	92.86	53.28	59.96	92.39	92.81	30.19	33.27	93.16	93.78	31.25	33.22
1800	20	4	96.67	97.02	55.78	63.72	96.43	96.92	32.52	35.17	96.74	97.04	31.39	32.83
1800	25	3.5	96.25	96.84	57.59	64.87	96.61	97.13	34.02	37.82	97.17	97.36	33.99	39.96
1800	25	3.75	99.11	99.25	60.44	68.52	99.08	99.19	34.91	41.39	99.25	99.36	35.46	42.31
1800	25	4	100	100	56.79	66.96	99.81	99.90	32.43	36.27	99.84	99.92	33.34	35.43

Table 17 Results of ZDS_2500 dataset between MCLP_CRO and Atta_GA (Atta et al. 2018)

n	p	s	Atta_GA				MCLP_CRO without LM-GM				MCLP_CRO with LM-GM			
			Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)	Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)	Avg Cov (%)	Max Cov (%)	Avg Time (s)	Max Time (s)
2500	15	3.5	69.47	69.64	58.92	63.87	68.79	69.52	28.26	29.94	69.03	70.06	28.23	30.69
2500	15	3.75	77.08	77.42	60.12	68.43	76.81	77.46	32.21	33.39	77.05	77.98	31.27	33.32
2500	15	4	83.52	83.96	62.91	71.26	84.12	84.76	35.49	39.14	84.98	85.72	35.64	38.95
2500	20	3.5	85.49	86.77	64.19	70.42	85.41	85.79	33.86	36.97	86.17	86.82	35.23	37.27
2500	20	3.75	91.62	91.98	62.68	71.43	91.31	91.84	33.71	37.23	92.37	92.90	32.24	35.88
2500	20	4	96.75	96.82	63.94	68.37	96.32	96.87	29.90	31.18	96.54	96.90	30.89	32.65
2500	25	3.5	95.72	96.02	68.22	74.34	95.76	96.07	34.21	39.05	95.84	96.09	33.26	37.25
2500	25	3.75	99.01	99.42	66.84	73.56	98.84	99.46	33.16	37.43	99.47	99.53	32.45	39.76
2500	25	4	99.93	99.99	64.99	73.57	99.64	99.81	37.27	41.25	99.88	99.94	34.59	36.47

Table 18 Comparison in average time improvement for the datasets

Dataset	No. of instances	Avg_T Atta_GA (s)	Avg_T MCLP_CRO (s)	Avg_T improvement (s)	Improvement in (%)
SJC_324	10	4.32	1.32	3.00	69.44
SJC_402	11	7.92	1.47	6.45	81.44
SJC_500	15	9.15	3.13	6.02	65.79
SJC_708	21	17.38	4.72	12.66	72.84
SJC_818	26	18.72	5.73	12.99	69.39
B_700	9	43.84	18.59	25.25	57.59
B_900	9	47.07	21.67	25.40	53.96
ZDS_1800	9	54.45	31.82	22.63	41.56
ZDS_2500	9	63.64	35.80	27.84	43.75

rate programs were used to create these two random datasets. Tables 16 and 17 show the comparison results for ZDS_1800 and ZDS_2500 datasets between MCLP_CRO and Atta_GA (Atta et al. 2018). These two datasets are tested with 3 different values of S . These values are 3.5, 3.75 and 4. The number of facilities to be installed for these two datasets are 15, 20 and 25. There are nine instances for these two datasets. For eight instances out of nine (ZDS_1800 dataset), we have obtained better results with respect to percentage of coverage (both average and maximum) compared to Atta_GA (Atta et al. 2018). However, the computational time (both average and maximum) of each of the nine instances is less than that of Atta_GA (ZDS_1800 dataset). For five instances out of nine (ZDS_2500 dataset), we have got better results with respect to percentage of average coverage compared to Atta_GA (Atta et al. 2018). With respect to percentage of max coverage, eight instances out of nine we have obtained better results. However, the computational time (both average and maximum) of each of the nine instances is less than that of Atta_GA for ZDS_2500 dataset.

The two random datasets (ZDS_1800, ZDS_2500) were used by Atta et al. (2018) and Zarandi et al. (2011) for solving the MCLP. Zarandi et al. (2011) tested their method with these two datasets having the distance graph for the datasets those are created randomly in the range [0, 30] in uniform distribution, and the demand values are also created randomly from a uniform distribution in the range [0, 100]. Zarandi et al. (2011) made the first attempt to use these specific types of random datasets to deal with MCLP using a well-known metaheuristic named genetic algorithm. The great achievement of their proposed method is its runtime. Later, another metaheuristic-based approach was proposed by Atta et al. (2018), which outperform the previous method in almost all the dimensions incorporating a local refinement strategy. In this paper, we have also used these two random datasets, and we have compared our proposed method with the method proposed by Atta et al. (2018). The eighth and

ninth rows of Tables 18 and 19 show the comparison between our proposed method and Atta_GA (Atta et al. 2018) with respect to average computational time and percentage of coverage, respectively. The average computational time for B_1800 dataset is 170.33 s in Zarandi et al. (2011), 54.45 s in Atta et al. (2018) and our proposed method obtained 31.82 s which is more efficient than the previous two methods.

5.3 Results summary with details of the datasets

The proposed MCLP_CRO method gives solution in less computational time for both small and large datasets. In Table 18, we have shown the comparison in average time improvement for the all the test datasets. Here, Avg_T denotes average computational time of a dataset. Our method improves average computational time about 81.44% for SJC_402 dataset, which is the highest compared to all nine datasets, whereas the lowest improvement is 41.56% which is for ZDS_1800 dataset. For small scale datasets, average time improvement is higher (more than 50%). On the contrary, the average time improvement is lower (around 42%) for large-scale datasets.

The proposed MCLP_CRO method works very well with respect to computational time in all the test instances of all the datasets (see Table 18), and the improvement in time is very significant, and in fact the lowest improvement is 41.56%. Table 19 shows the significance of the average percentage of coverage of the proposed method where Avg_E represents the average gap of percentage of coverage from the best known percentage of coverage. For the small-scale datasets (SJC_324, SJC_402 and B_700), the gap is zero which is pretty good. The proposed method deals with the large-scale datasets with average gap of percentage of coverage with values 0.07, 0.07, 0.07, 0.16 and 0.05 for SJC_708, SJC_818, B_900, ZDS_1800 and ZDS_2500 datasets, respectively, which is also very less. Apart from this, only SJC_500 dataset contains average gap of percent-

Table 19 Results of summary in average percentage of coverage for the datasets

Dataset	No. of instances	Best result found instances	Avg_E (% of Cov)
SJC_324	10	10	0
SJC_402	11	11	0
SJC_500	15	13	0.49
SJC_708	21	20	0.07
SJC_818	26	22	0.07
B_700	9	9	0
B_900	9	8	0.07
ZDS_1800	9	8	0.16
ZDS_2500	9	8	0.05
Total	119	109	Avg = 0.10

age of coverage value 0.49, which is the worst compared to other seven datasets while ensuring 65.79% improvement in average computational time. The proposed MCLP_CRO method gives best percentage of coverage results in 91.60% of instances, while rest of the 8.40% of instances produce results with average error value 0.10. Nevertheless, the proposed method works very well with respect to computational time in all the test instances of all the datasets (see Table 18) and the improvement in time is very significant, in fact the lowest improvement is 41.56% compared to the state-of-the-art method.

5.4 Statistical test (Wilcoxon signed-rank test)

Here Wilcoxon signed-rank test is used for significance test; a nonparametric statistical experiment is carried out between two paired set of values to determine whether there is any statistical relationship between them or not. We have examined the test between MCLP_CRO and Atta_GA (Atta et al. 2018) to observe the statistical significance. An open-source online statistical calculator is used (<https://www.socscistatistics.com/tests/signedranks/default2.aspx>) to run the test. We have performed the test for both real-world and random instances between MCLP_CRO and Atta_GA, and the results are shown in Tables 20 and 21, respectively. The environment of the test: hypothesis is 2-tailed and the level of significance is 0.05.

The test result highlights that MCLP_CRO is significant at $P < 0.05$ in both real-world and random instances. The results of the real-world instances (Table 20) indicate that the value of W is 26.5. The critical value for W at $N = 18$ ($P < 0.05$) is 40. The Z -value cannot be used to evaluate the hypothesis for real-world instances as the size of N is 18 which means there are less samples to form a pure normal distribution. The test requires at least 20 samples to form a

Table 20 Statistical test for real-world instances between MCLP_CRO and Atta_GA (Atta et al. 2018)

Testing parameter	Value
W-value	26.5
Difference of mean	68.22
Summation of positive ranks	26.5
Summation of negative ranks	144.5
Z-value	-2.5695
Mean (W)	85.5
Standard deviation (W)	22.96
P-value	0.01016
Size of sample (N)	18
Result	Significant

Table 21 Statistical test for random instances between MCLP_CRO and Atta_GA (Atta et al. 2018)

Testing parameter	Value
W-value	18.5
Difference of mean	10.09
Summation of positive ranks	18.5
Summation of negative ranks	647.5
Z-value	-4.941
Mean (W)	333
Standard deviation (W)	63.65
P-value	0.00001
Size of sample (N)	36
Result	Significant

pure normal distribution. So, W -value is used for evaluating the hypothesis of real-world instances.

The result of the random instances (Table 21) shows that the value of W is 18.5. The distribution is approximately normal. Therefore, the Z -value can be used to evaluate the hypothesis for random instances.

6 Conclusion

The maximal covering location problem (MCLP) deals with the problem of finding an optimal placement of a given fixed number of facilities on a network in such a way that the total demands of the attended population are maximized. MCLP is a well-known constraint satisfaction problem, which is also discussed in location science. In this article, a chemical reaction optimization (CRO)-based method is proposed to solve the maximal covering location problem (MCLP); hence, the proposed method is named as MCLP_CRO. The goal of this paper is to solve MCLP using CRO and achieve the best percentage of coverage results in least computational time in

both benchmark and random types of datasets and propose an efficient way of solving the well-known location covering problem, MCLP, and also observe the statistical significance of the proposed method with other existing methods. Designing the four fundamental operators of CRO was a challenging task to solve the problem. We have performed parameter tuning to initialize the CRO parameters. Sometimes the solutions generated by the method get trapped into local maxima, to get rid of this problem we have designed a repair operator (LM-GM) which converts the local maximum into global maximum and obtain the best results. The proposed algorithm was tested for both small and large scales of instances of datasets, which include benchmark (5 datasets) as well as random ones (4 datasets). The proposed MCLP_CRO method gives best percentage of coverage results in 91.60% of instances, and for the remaining 8.40% of instances it produces results with average error value 0.10% which is very close to the optimal value. Nevertheless, the proposed method works very well with respect to computational time in all the test instances (100%) of all the datasets and the improvement in computational time is very significant; in fact, the lowest improvement is 41.56% for ZDS_1800 dataset, while the highest improvement is 81.44% which is for SJC_402 dataset. For small-scale datasets, average time improvement is higher (more than 50%). On the contrary, the average time improvement is lower (around 42%) for large-scale datasets. In a nutshell, the proposed method gives solution in less computational time for both small and large datasets compared to the state-of-the-art algorithm (Atta_GA). An statistical test (Wilcoxon signed rank test) was performed on the results of the method to observe the statistical significance with 2-tailed hypothesis and 0.05 level of significance. The statistical test shows that the proposed method is significant in both real-world and random instances. *W*-value (26.5) is used for evaluating the hypothesis of real-world instances and *Z*-value (−4.941) can be used to evaluate the hypothesis for random instances. So, the proposed method is an efficient one from the perspective of computational time. Nevertheless, it has a drawback that it cannot produce optimal percentage of coverage results for some test instances though the difference is very less, only 0.10%. In future, the traditional MCLP can be converted to dynamic large scales of MCLP to find dynamic solutions in different resource constraint aspects. A fuzzy CRO version of MCLP can be deployed in future. Multiple metaheuristics could be used at the same time to solve MCLP for further betterment in future. Another avenue for future research could be to assess the performance of CRO for other variants of covering location problem.

Funding The authors (authors including corresponding author) declare that they have no funding for this work.

Data availability The data generated and analyzed during the study are included.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest The authors declare that they have no conflict of interest.

References

- Al-Janabi S, Alkaim A (2022) A novel optimization algorithm (Lion-AYAD) to find optimal DNA protein synthesis. *Egypt Inform J* 23(2):271–290. <https://doi.org/10.1016/j.eij.2022.01.004>
- Atta S, Mahapatra PRS, Mukhopadhyay A (2018) Solving maximal covering location problem using genetic algorithm with local refinement. *Soft Comput* 22:3891–3906. <https://doi.org/10.1007/s00500-017-2598-3>
- Bagherinejad J, Bashiri M, Nikzad H (2018) General form of a cooperative gradual maximal covering location problem. *J Indust Eng Int* 14:241–253. <https://doi.org/10.1007/s40092-017-0219-5>
- Bargaoui H, Driss OB, Ghédira K (2017) A novel chemical reaction optimization for the distributed permutation flowshop scheduling problem with makespan criterion. *Comput Indust Eng* 111:239–250. <https://doi.org/10.1016/j.cie.2017.07.020>
- Barham R, Sharieh A, Sliet A (2016) Chemical reaction optimization for max flow problem. *IJACSA Int J Adv Comput Sci Appl* 7(8):189–196
- Berman O, Krass D (2002) The generalized maximal covering location problem. *Comput Oper Res* 29(6):563–581. [https://doi.org/10.1016/S0305-0548\(01\)00079-X](https://doi.org/10.1016/S0305-0548(01)00079-X)
- Bhoi NK, Singh H, Pratap S, Jain PK (2022) Chemical reaction optimization algorithm for machining parameter of abrasive water jet cutting. *OPSEARCH* 59:350–363. <https://doi.org/10.1007/s12597-021-00547-z>
- Crama Y, Schyns M (2003) Simulated annealing for complex portfolio selection problems. *Eur J Oper Res* 150(3):546–571. [https://doi.org/10.1016/S0377-2217\(02\)00784-1](https://doi.org/10.1016/S0377-2217(02)00784-1)
- Davari S, Zarandi MHF, Hemmati A (2011) Maximal covering location problem (MCLP) with fuzzy travel times. *Expert Syst Appl* 38(12):14535–14541. <https://doi.org/10.1016/j.eswa.2011.05.031>
- Hajipour V, Fattahi P, Bagheri H et al (2022) Dynamic maximal covering location problem for fire stations under uncertainty: soft-computing approaches. *Int J Syst Assur Eng Manag* 13:90–112. <https://doi.org/10.1007/s13198-021-01109-8>
- Islam MR, Saifullah CMK, Asha ZT, Ahamed R (2019) Chemical reaction optimization for solving longest common subsequence problem for multiple string. *Soft Comput* 23:5485–5509. <https://doi.org/10.1007/s00500-018-3200-3>
- Islam MR, Smrity RA, Chatterjee S, Mahmud MR (2020) Optimization of protein folding using chemical reaction optimization in HP cubic lattice model. *Neural Comput Appl* 32:3117–3134. <https://doi.org/10.1007/s00521-019-04447-8>
- Islam MR, Protik P, Das S, Boni PK (2021) Mobile robot path planning with obstacle avoidance using chemical reaction optimization. *Soft Comput* 25:6283–6310. <https://doi.org/10.1007/s00500-021-05615-6>
- Islam MR, Islam MS, Sakeef N (2021) RNA secondary structure prediction with pseudoknots using chemical reaction optimization

- algorithm. *IEEE/ACM Trans Comput Biol Bioinform* 18(3):1195–1207. <https://doi.org/10.1109/TCBB.2019.2936570>
- Janabi SA, Alkaim A, Taleby GA (2023) Triple steps for verifying chemical reaction based on deep whale optimization algorithm (VCR-WOA). *Recurrent neural networks*. CRC Press, Boca Raton, pp. 183–201. <https://doi.org/10.1201/9781003307822-14>
- Jin Xu, Lam Albert Y.S., Li Victor O.K (2011) Chemical reaction optimization for task scheduling in grid computing. *IEEE Trans Parallel Distrib Syst* 22(10):1624–1631. <https://doi.org/10.1109/TPDS.2011.35>
- Kabir R, Islam R (2019) Chemical reaction optimization for RNA structure prediction. *Appl Intell* 49(2):352–375. <https://doi.org/10.1007/s10489-018-1281-4>
- Kadhuim ZA, Janabi SA (2023) Codon-mRNA prediction using deep optimal neurocomputing technique (DLSTM-DSN-WOA) and multivariate analysis. *Results Eng* 17:100847. <https://doi.org/10.1016/j.rineng.2022.100847>
- Lam AYS, Li VOK (2012) Chemical reaction optimization: a tutorial. *Math Comput Modell* 4(1):3–17. <https://doi.org/10.1007/s12293-012-0075-1>
- Li JQ, Pan QK (2012) Chemical-reaction optimization for flexible job-shop scheduling problems with maintenance activity. *Appl Soft Comput* 12(9):2896–2912. <https://doi.org/10.1016/j.asoc.2012.04.012>
- Li H, Mukhopadhyay SK, Wu J, Zhou L, Du Z (2020) Balanced maximal covering location problem and its application in bike-sharing. *Int J Prod Econ* 223:107513. <https://doi.org/10.1016/j.ijpe.2019.09.034>
- Lorena LAN, Pereira MA (2002) A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillman's edition. *Int J Indust Eng* 9(1):57–67
- Mahmud MR, Pritom RM, Islam MR (2017) Optimization of collaborative transportation scheduling in supply chain management with TPL using chemical reaction optimization. 2017 20th international conference of computer and information technology (ICCIT), Dhaka, pp 1–6. <https://doi.org/10.1109/ICCITECHN.2017.8281767>
- Máximo VR, Nascimento MCV, Carvalho AC (2017) Intelligent-guided adaptive search for the maximum covering location problem. *Comput Oper Res* 78:129–137. <https://doi.org/10.1016/j.cor.2016.08.018>
- Megiddo N, Zemel E, Hakimi SL (1983) The maximum coverage location problem. *SIAM J Algebr Discret Methods* 4(2):253–261. <https://doi.org/10.1137/0604028>
- Saha SK, Islam MR, Hasan M (2020) DNA motif discovery using chemical reaction optimization. *Evol Intell*. <https://doi.org/10.1007/s12065-020-00444-2>
- Saifullah CMK, Islam MR (2016) Solving shortest common supersequence problem using chemical reaction optimization. 2016 5th international conference on informatics, electronics and vision (ICIEV), Dhaka, pp 50–55. <https://doi.org/10.1109/ICIEV.2016.7760187>
- Salman S, Alaswad S (2022) Designing reduced congestion road networks via an elitist adaptive chemical reaction optimization. *Comput Indust Eng* 163:107788. <https://doi.org/10.1016/j.cie.2021.107788>
- Syah R, Guerrero JWG, Poltarykhin AL, Suksatan W et al (2023) Developed teamwork optimizer for model parameter estimation of the proton exchange membrane fuel cell. *Energy Rep* 8:10776–10785. <https://doi.org/10.1016/j.egy.2022.08.177>
- Truong TK, Li K, Xu Y (2013) Chemical reaction optimization with greedy strategy for the 0–1 knapsack problem. *Appl Soft Comput* 13(4):1774–1780. <https://doi.org/10.1016/j.asoc.2012.11.048>
- Xu J, Lam AYS, Li VOK (2010) Parallel chemical reaction optimization for the quadratic assignment problem. The 2010 world congress in computer science, computer engineering, and applied computing (Worldcomp 2010), Las Vegas, NV, pp. 12–15 July 2010. In Conference proceedings, 2010
- Zain AM, Yousif A (2020) Chemical reaction optimization (CRO) for cloud job scheduling. *SN Appl Sci* 2(1):23. <https://doi.org/10.1007/s42452-019-1758-8>
- Zarandi MHF, Davari S, Sisakht SAH (2011) The large scale maximal covering location problem. *Sci Iran* 18(6):1564–1570. <https://doi.org/10.1016/j.scient.2011.11.008>
- Zarandi MHF, Davari S, Sisakht SAH (2013) The large-scale dynamic maximal covering location problem. *Math Comput Modell* 57(13):710–719

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.