Solving Maximal Covering Location Problem Using Chemical Reaction Optimization

Md. Shymon Islam(✉), Md. Rafiqul Islam, and Humaira Islam

Computer Science and Engineering Discipline, Khulna University,
Khulna 9208, Bangladesh
shymum1702@cseku.ac.bd

Abstract. The Maximal covering location problem (MCLP) works with a given number of nodes in a network, each node has a demand value and is provided with a fixed number of facilities. Here the target is to maximize the total demands within some constraints. In this article, we have proposed a metaheuristic algorithm based on chemical reaction optimization to solve this problem. We have used two data sets to measure the performance of our proposed method. The proposed method gives better results in almost all the test cases in terms of percentage of coverage as well as computational time.

Keywords: Chemical reaction optimization · Demand maximization · Maximal covering location problem (MCLP) · Meta-heuristic · NP hard

1 Introduction

The covering location problem (CLP) is a well-known type of location problem discussed in location science [1]. The maximal covering location problem (MCLP) is a specific type of CLP that aims at finding an optimal placement of a given number of facilities to each customer or node on a network in such a way that the total demands covered by the served population is maximized. MCLP maintains the constraint that a customer is covered by a facility if it falls within the given constant service area or coverage area [2]. 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 [3]. Maximal covering location problem is also a constraint satisfaction problem that is very useful in locating objective areas to be served in different context. The MCLP has importance not only in private sectors but also in public sectors. Placement of plant, warehouse, telecommunication antennas etc. are the examples of private sectors where MCLP can be applied to find an optimal placement structure. Locating schools, bus-stops, ambulances, parks etc. examples of some public sectors where MCLP is applicable. MCLP is greatly applicable in designing network with constraints. The applications areas of MCLP have motivated us to solve the problem efficiently.

K. R. Venugopal et al. (Eds.): ICInPro 2021, CCIS 1483, pp. 424–437, 2021.
https://doi.org/10.1007/978-3-030-91244-4_34

There are several techniques for solving the maximal covering location problem. Many algorithms were proposed to solve the MCLP by researchers such as simulated annealing ([1] and [3]), lagrangean/surrogate heuristic [4], genetic algorithm [5], greedy heuristic ([6] and [7]), tabu search heuristic [8]. Although many algorithms were proposed and applied to solve MCLP, there is no general algorithm that can give always optimal results. This is the scope to work with MCLP. We have solved the MCLP by using chemical reaction optimization (CRO) algorithm. CRO is a nature based meta-heuristic approach. In recent years, CRO has successfully solved so many optimization problems such as shortest common super-sequence problem [9], RNA structure prediction [10], transportation scheduling in supply chain management with TPL [11], RNA secondary structure prediction with pseudoknots [12], optimization of protein folding [13], flexible job-shop scheduling problems with maintenance activity [14], the distributed permutation flow-shop scheduling problem with makespan criterion [15], the cloud job scheduling [16] etc. with better results than the other existing meta-heuristics algorithms. Due to the performance of CRO, we have chosen this algorithm. We are optimistic that CRO can find the optimal solution in less computational time. The contributions of the proposed work: redesigned the four reaction operators and obtained best results in both the datasets in less computational time compared to the state of the art algorithm (Atta_GA [2]). In this article, we have demonstrated the basic ideas of MCLP in Sect. 1. Section 2 describes the problem statement, Sect. 3 is for related work, Sect. 4 demonstrates our proposed method for solving MCLP using chemical reaction optimization and Sect. 5 concludes the work.

2 Problem Statement

The MCLP does the business with the problem of searching an optimal placement of a given number of facilities on a network in such a way that the total demands of the attended population is maximized [5]. MCLP works with a given number of customers (nodes) in a network. Each customer (node) has three basic properties: demand value, x and y coordinate values. The two coordinate values (x, y) define the position of a customer (node) in the network. Each customer (node) is provided with a fixed number of facilities. Each facility has a service area called the coverage area which is circular in shape and it remains constant throughout the whole process. A facility can be provided to a customer (node) if the Euclidean distance between the concerned node and the facility node is less than the constant service area. The MCLP is NP-hard [17]. The solution 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 conditions. That's why meta-heuristic approaches are applicable to determine efficient solution for this problem.

The mathematical representation of the objective function of MCLP is given by the equation as follows:

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

Expression (1) is taken from [2]. Here x and d_x denotes the index of demand nodes and the demand value at node (customer) x respectively. And s_x is a binary decision variable which becomes 1 when node x is selected and 0 otherwise. The target of the objective function (1) is to maximize the total demands covered by the selected customers. The objective function have two constraints. The first constraint is that each customer is provided with exactly a given fixed number of facilities. And the second one is that a customer (node) is covered or selected when there exists at least one facility within the coverage area of the node.

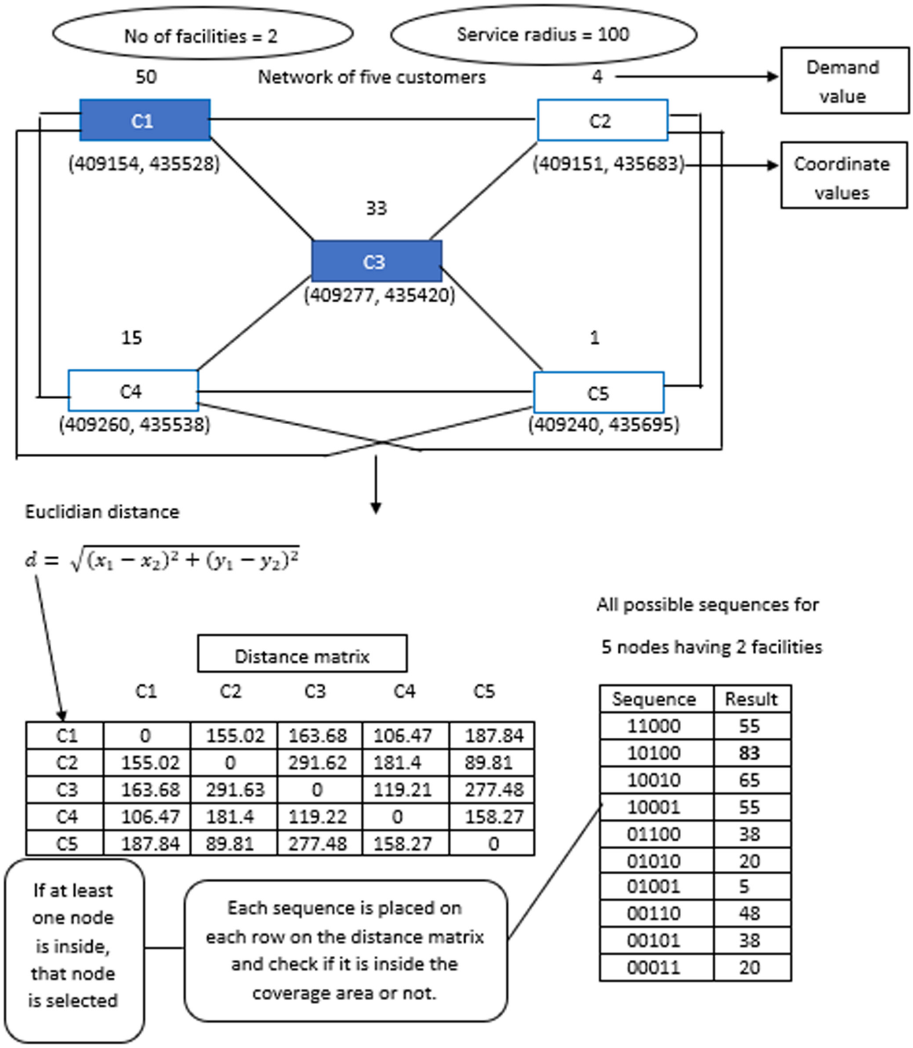
A sample MCLP example is given in Fig. 1. There are five nodes labeled as C1, C2, C3, C4 and C5. The corresponding demand and coordinate values of each node are presented in the upper and lower portion of each node respectively. The service distance for this network is 100, and the number of facilities to be installed is 2. This is a fully connected graph. For solving this MCLP graph, firstly a distance matrix needs to be created by using the Euclidian distance formula. Then All the possible solutions for five customers having 2 facilities needs to be generated. The total number of sequences is $\frac{5_{p2}}{2!}$ that is 10. Each sequence needs to be traversed in the whole distance matrix and finds the result. So, ten sequences must have ten results, among them the highest value will be the output. In the sample example, the highest value is 83 that is generated for sequence 10100. The final percentage of coverage is calculated by the ratio between sum of all demand values of the customers and the generated result. In Fig. 1, node C1 and C3 are selected because it maximizes the demand values of attended nodes. So, C1 and C3 are colored. Therefore, the estimated result for the graph in Fig. 1 is 83. So, the percentage of coverage is $\frac{83}{103} \times 100$ that is 80.58%.

3 Related Work

Some of the existing algorithms of MCLP are described here.

In 2018, Atta S. et al. proposed an algorithm for solving the maximal covering location problem (MCLP) using genetic algorithm with local refinement [2]. A binary array representation is used as the encoding scheme. The authors designed the selection, crossover and mutation operators respectively along with a local refinement procedure and elitism. Each chromosome goes through to these operators and tries to find out better solutions. They compared their proposed algorithm with other approaches with respect to percentage of coverage and computational time. Most of the cases, their proposed algorithm gives a fair better results in both percentage of coverage and computational time.

Maximo V.R. et al. proposed an intelligent guided adaptive search method for solving the maximum covering location problem (MCLP) in 2016 [8].


No of facilities = 2
Service radius = 100
Network of five customers
Demand value
Coordinate values
50
C1
(409154, 435528)
4
C2
(409151, 435683)
33
C3
(409277, 435420)
15
C4
(409260, 435538)
1
C5
(409240, 435695)
Euclidian distance
Distance matrix
All possible sequences for
5 nodes having 2 facilities
If at least one node is inside, that node is selected
Each sequence is placed on each row on the distance matrix and check if it is inside the coverage area or not.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

	C1	C2	C3	C4	C5
C1	0	155.02	163.68	106.47	187.84
C2	155.02	0	291.62	181.4	89.81
C3	163.68	291.63	0	119.21	277.48
C4	106.47	181.4	119.22	0	158.27
C5	187.84	89.81	277.48	158.27	0

Sequence	Result
11000	55
10100	**83**
10010	65
10001	55
01100	38
01010	20
01001	5
00110	48
00101	38
00011	20

Fig. 1. An example of MCLP

The proposed method by this paper is named as Intelligent-Guided Adaptive Search (IGAS) and it actually follows the Greedy randomized search procedure (GRASP). Artificial neural network is used to build the construction phase of IGAS. This method is specialized for large-sized instances (more than 3000 nodes). An unsupervised machine learning algorithm, GNG (Neural Gas) is used to keep track of the best solutions found in the current iteration. So, the algorithm tries to take decision in such a way that, it can traverse through a promising branch.

In 2013, Zarandi M.H.F. et al. proposed an approach for solving the large-scale dynamic maximal covering location problem [1]. This paper proposed a multi period version of MCLP called the dynamic MCLP (DMCLP). They used simulated annealing (SA) algorithm to solve the problem. Neighborhood search structure (NSS) is used to search for better solutions.

Davari S. et al. proposed a method for solving the maximal covering location problem (MCLP) with fuzzy travel times in 2011 [3]. They solved the problem by using a hybrid intelligent algorithm where they incorporated both fuzzy simulation (FS) and simulated annealing (SA) techniques. At the beginning, SA generates the initial population with several solutions and each solution is generated either randomly or by applying some predefined assumptions. The iteration process starts with the initial population, and in each iteration the proposed algorithm tries to find further better solutions than the existing ones.

In 2011, Zarandi M.H.F. et al. proposed a method for solving the large-scale maximal covering location problem [5]. Their proposed method is applicable for upto 2500 nodes. This method applied a binary vector representation to complete the chromosome encoding. In each generation the selection, crossover and mutation operators are carried out to search for better solutions than the existing ones. The roulette wheel selection (RWS) is used to design the selection operator. Crossover is carried out between two chromosomes selected by the selection operator. Finally, the mutation is carried out to perform the diversification of the solution space.

4 Proposed Method for Solving MCLP Using Chemical Reaction Optimization

Lam A.Y.S. et al. proposed an algorithm for optimization problem named Chemical Reaction Optimization (CRO) [18]. The working principle of CRO actually follows the two laws of thermodynamics. The first law (law of conservation) states that energy can not be created or destroyed. Energy can only be transformed from one form to another. Hence, the total amount of energy remains constant. This can be represented by the equation as follows:

$$\sum_{x=1}^{Popsize(t)} (PE_x(t) + KE_x(t) + buffer(t) = C \tag{2}$$

In Eq. (2), $PE_x(t)$ and $PE_x(t)$ represents the potential energy (PE) and kinetic energy (KE) of molecule x respectively at any time t. $Popsize(t)$ is the total number of molecules, and $buffer(t)$ is the energy in the central buffer at time t. And the total constant energy is denoted by C. The value of C proves that the conservation of energy is maintained in CRO. The second law of thermodynamics ensures transformation of energy among the molecules (PE is converted to KE during iteration stage). CRO follows these two rules to come up with a better solution for optimization problems.

CRO is a population-based metaheuristic. Basically, CRO performs three basic stages. These are the initialization stage, iterations stage and the final stage. In the initialization stage, CRO initializes several attributes (initial parameters) and creates the initial population. Each molecule have several attributes or parameters. 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 generates the initial population with popsize (total number of molecules). The second stage is the iterations stage. CRO has four elementary reaction operators. These are the on-wall ineffective collision, intermolecular ineffective collision, synthesis and decomposition. The on-wall ineffective collision and the intermolecular ineffective collision ensures intensification (local search). One the contrary, the synthesis and decomposition operator ensures diversification (global search). These four reaction operators make CRO a better approach for optimization problems compared to other metaheuristics because of its capability of searching. CRO increases or decreases the total number of molecules in each iteration according to the type of operator activated. The third stage is the final stage of CRO. If any of the stopping criteria is met during iterations or the limit of iteration exceeds then CRO will go to the final stage and show necessary outputs.

4.1 Basic Structure of Proposed Algorithm

$MCLP\_CRO$ is our proposed method. The initial parameters of CRO is initialized with proper values, then create the initial population randomly. By passing these initial parameters and initial population to the function MCLP_CRO, perform the iterations step. After the termination of MCLP_CRO, measure the outputs. The outputs are saved into two variables named *result* and *time*. The *result* represents the percentage of coverage value and *time* represents the computational time of the proposed method.

4.2 Solution Generation and Initialization

CRO is a population-based metaheuristic. A single unit from the whole population is called a molecule. In the iterations stage, one of the operators is manipulated to come up with new molecules with better objective function value. Here the interesting thing is that, the total number of molecules in each iteration does not remain constant, rather it varries from iteration to iteration. All the molecules with the popsize creates the whole population. Each molecule have several parameters. Some of the initial parameters of CRO are given in Table 1.

Population Generation. The initial population is generated on the basis of random selection. Let there are m customers where f facilities to be located. Each molecule of the initial population is generated by selecting f random indices from the set of customers or nodes $\{1, 2, 3, ..., m\}$. Here the value of f is less than m. And each f is generated randomly (random function mod popsize). And by

Table 1. Initial parameters

Symbol	Algorithmic definition
PopSize	Population zize (solution space)
KELossRate	Kinetic energy (KE) loss rate
Molecoll	Decision parameter (unimolecular or bimolecular) of CRO
Buffer	Initial energy in the surroundings
InitialKE	Initial kinetic energy
α and β	Threshold values of CRO
NumHit	Total number of hits a molecule has taken
Minstruct	Structure with minimum potential energy
MinPE	The potential energy when a molecule has minstruct
MinHit	The number of hits when a molecule has minstruct

looping through 0 to popsize, all the molecules are generated accordingly. Thus the initial population is created.

Solution Representation. A solution for MCLP is a set of f potential locations those needs to be chosen from the set of m customers. Let $m = 10$ and $f = 3$. Here, m and f are both are represented by one dimensional array. And initially all the values are 0. Let the randomly selected indices for the potential facility sites are $[3, 5, 9]$. Table 2 represents the indexed f array.

Table 2. Indexed f array

Index	1	2	3
Value	3	5	9

Table 3. Solution representation

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

A binary vector representation method is used for solution representation (see Table 3). In the solution, the selected indices of f array are represented by 1 and rest of indices are represented by 0.

4.3 Operator Design

CRO has four reaction operators. These are the on-wall ineffective collision, intermolecular ineffective collision, decomposition and synthesis. These operators are preformed selectively in iterations stage of CRO.

On Wall Ineffective Collision. When one molecule collides with the wall of a container, then the internal structure of the molecule changes. Here molecule m produces a new molecule m'

$$m \rightarrow m'$$

Let $m = 10$ and $f = 3$. So, each customer is provided with 3 facilities. The mechanism for on wall ineffective collision is very simple. Randomly select one or more indices from molecule m and change it to create a new molecule m'. In Fig. 2 we can see the 7^{th} and 9^{th} indices of molecule m are changed to form new molecule m'.

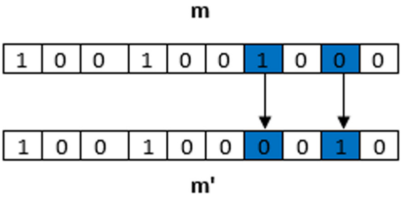


Fig. 2. On wall ineffective collision

Decomposition. In this elementary reaction, two new molecules are generated from a molecule. Two newly generated molecules bring diversity in their structure from the old molecule. Let, molecule m produces two new molecules m_1 and m_2.

$$m \rightarrow m_1 + m_2$$

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 m into two portions (see Fig. 3) using a random divider function. Then, the first and second portions of molecule m are copied to the beginning of molecule m_1 and m_2 respectively. The rest of the indices of molecule m_1 and m_2 are selected randomly using a random function generator.

Inter-molecular Ineffective Collision. In an inter-molecular ineffective collision, two molecules collide with each other. Let two molecules m_1 and m_2 collide with each other and produce two new molecules $m_1{}'$ and $m_2{}'$.

$$m_1 + m_2 \rightarrow m_1{}' + m_2{}'$$

This is much similar to On-wall ineffective collision except that the number of molecules is twice here. Molecule $m_1{}'$ is produced from molecule m_1 and molecule $m_2{}'$ is produced from m_2. The mechanism is same as on wall ineffective collision and it is shown in Fig. 4. Several indices are changed to form a new molecule.

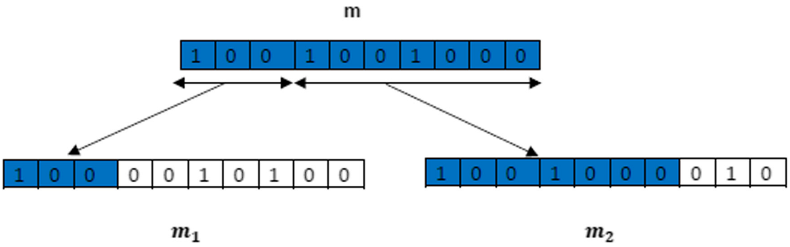


Fig. 3. Decomposition

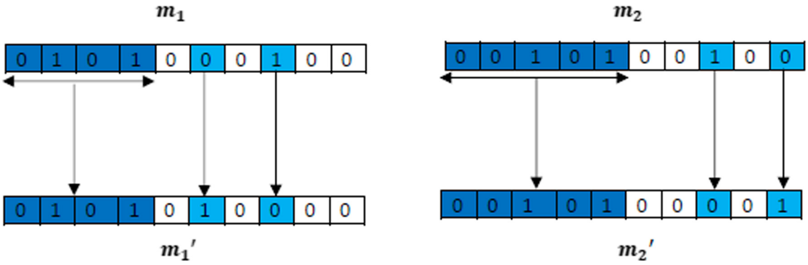


Fig. 4. Inter-molecular ineffective collision

Synthesis. Synthesis operator consolidates two molecules to form a new molecule. It is a reverse procedure of decomposition. Let m_1 and m_2 be two molecules. After the collision, molecule m is created.

$$m_1 + m_2 \to m$$

Synthesis operator performs diversification to traverse the global solution space and tries to find the optimal solution. One point crossover mechanism (see Fig. 5) is used for synthesis. The first few indices of molecule m are copied from molecule m_1 and rest of the indices are copied from molecule m_2.

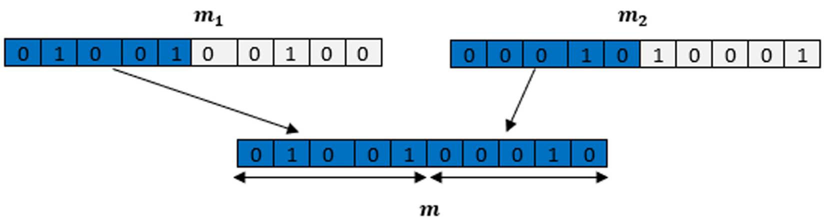


Fig. 5. Synthesis

4.4 Flowchart of Proposed Algorithm

In Fig. 6 we have shown the flowchart of the proposed method. Firstly generate the initial population with a random function generator. Then check whether the reaction is unimolecular or intermolecular. If intermolecular reaction occurs, then either do inter-molecular ineffective collision or synthesis by checking which one is appropriate. Otherwise perform on-wall ineffective collision or decomposition in same manner. Then check if a max point is found or not. If any max point is found then check for 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 unimolecular or inter-molecular. Do the same again until get a max point or reach the iteration limit.

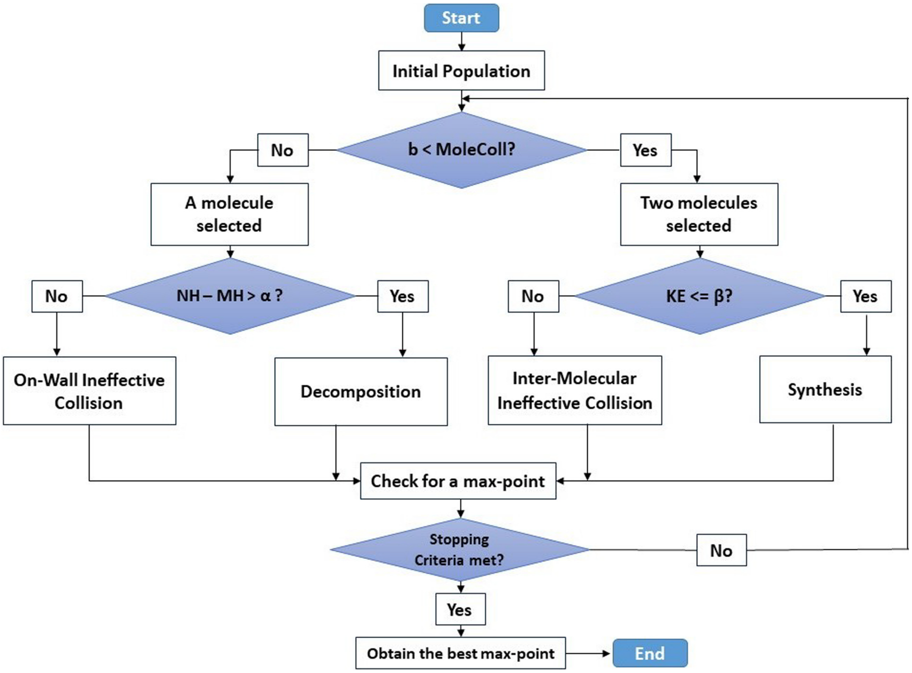


Fig. 6. Flowchart of proposed algorithm (MCLP_CRO)

4.5 Experimental Results and Comparisons

We have taken two real world datasets to measure the performance of our algorithm. These datasets are named as SJC_324 and SJC_402 where 324 and 402 are the total number of nodes in the datasets respectively. The datasets were collected from *L.A.N.Lorena-instancias* - *INPE* whose link is available in the

Table 4. Comparison of SJC_324 dataset between MCLP_CRO and Atta_GA [2]

n	p	s	Known best	Atta_GA		MCLP_CRO	
			% of Cov [4]	% of Cov	Time (s)	% of Cov	Time (s)
324	1	800	44.94	44.94	2.42	**44.94**	**1.45**
324	2	800	72.33	72.33	3.79	**72.33**	**1.41**
324	3	800	95.49	95.49	5.49	**95.49**	**1.73**
324	4	800	99.62	99.62	5.07	**99.62**	**1.65**
324	5	800	100	100	6.45	99.80	**1.16**
324	1	1200	81.73	81.73	2.95	**81.73**	**1.47**
324	2	1200	95.08	95.08	4.25	**95.08**	**1.08**
324	3	1200	100	100	5.34	**100**	**1.16**
324	1	1600	99.76	99.76	3.51	**99.76**	**1.46**
324	2	1600	100	100	3.94	**100**	**1.27**

Table 5. Comparison of SJC_402 dataset between MCLP_CRO and Atta_GA [2]

n	p	s	Known best	Atta_GA		MCLP_CRO	
			% of Cov [4]	% of Cov	Time (s)	% of Cov	Time (s)
402	1	800	41.01	41.01	4.03	**41.01**	**1.47**
402	2	800	70.94	70.94	6.40	**70.94**	**1.34**
402	3	800	91.90	91.90	8.79	**91.90**	**1.33**
402	4	800	97.85	97.85	8.34	**97.96**	**1.38**
402	5	800	99.91	99.91	10.52	99.28	**1.55**
402	6	800	100	100	13.36	**100**	**1.73**
402	1	1200	66.36	66.36	5.18	**66.36**	**1.48**
402	2	1200	92.79	92.79	8.86	**92.79**	**1.39**
402	3	1200	100	100	8.75	**100**	**1.64**
402	1	1600	99.58	99.58	5.93	**99.58**	**1.51**
402	2	1600	100	100	6.92	**100**	**1.66**

website (http://www.lac.inpe.br/~lorena/instancias.html). These datasets were used for evaluating recent algorithms for solving maximal covering location problem ([2] and [4]).

We have implemented our proposed algorithm (MCLP_CRO) by using C++ language with device specifications: Processor- Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz 2.71 GHz, RAM - 8.00 GB (7.90 GB usable), System type - 64-bit operating system, x64-based processor, OS - Windows 10 Pro edition. We have initialized the CRO parameters as iteration = 150, PopSize = 10, KELoss-

Rate = 0.2, MoleColl = 0.4, InitialKE = 1000, $\alpha = 5$ and $\beta = 15,000$. We have recorded the input parameters as n, p and S, where n denotes the total number of customers or nodes in the network, p represents the total number of facilities to be installed and S is the constant service distance or service radius inside where the facilities can be provided to a node or customer. The $\%\ of\ Cov$ and $Time(s)$ denotes the percentage of coverage and the computational time respectively. The best known $\%\ of\ Cov$ values for both the datasets have been collected from [4]. For a fair comparison, we have implemented Atta_GA [2] also using C++ in the same machine with the same device specification as for MCLP_CRO. The comparison for SJC_324 dataset between MCLP_CRO and Atta_GA [2] is shown is Table 4. Both datasets are tested with 3 different values of S (service distance). 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 percentage of coverage and the computational time. The good results of our proposed method is highlighted by bold sign. For 9 instances out of 10, the proposed method gives better result than Atta_GA [2]. For test input ($n = 324$, $p = 5$ and $S = 800$), our method gave worse result than Atta_GA [2] with respect to percentage of coverage. The comparison for SJC_402 dataset between MCLP_CRO and Atta_GA [2] is shown is Table 5. There are 11 instances in SJC_324 dataset. For 10 instances out of 11, we have got good result compared to Atta_GA [2]. For test input ($n = 402$, $p = 5$ and $S = 800$), our method gave worse result than Atta_GA [2] with respect to percentage of coverage. To observe the efficiency of our proposed method, we have shown two graphs (Fig. 7 and Fig. 8) with service distance 800. Both graphs show that that Atta_GA [2] method takes more time than the proposed method in both datasets.

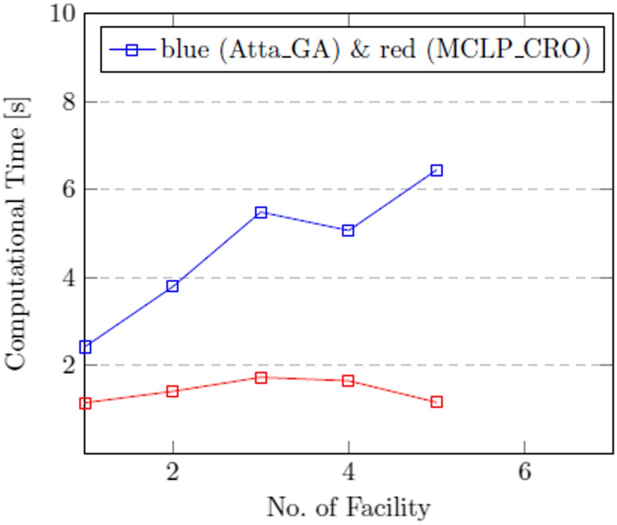


Fig. 7. Time comparison of SJC_324 dataset with $S = 800$

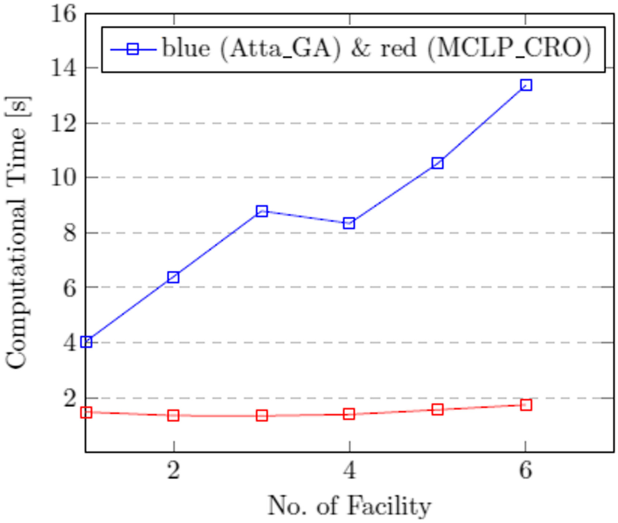


Fig. 8. Time Comparison Of Sjc_402 dataset with $S = 800$

5 Conclusion

In this article, we have proposed a CRO based method for solving the maximal covering location problem. Design the four reaction operators of CRO was a tough task. The proposed algorithm gives best results in almost all cases. We have obtained best results in less running time compared to the state-of-art algorithm (Atta_GA). The proposed algorithm was tested in small scale of MCLP. In the future, we will improve our algorithm further so that it can perform well on both small and large scales of MCLP.

References

1. Zarandi, M.H.F., Davari, S., Sisakht, S.A.H.: The large-scale dynamic maximal covering location problem. Math. Comput. Model. **57**(13), 710–719 (2013)
2. Atta, S., Sinha Mahapatra, P.R., Mukhopadhyay, A.: Solving maximal covering location problem using genetic algorithm with local refinement. Soft. Comput. **22**(12), 3891–3906 (2017). https://doi.org/10.1007/s00500-017-2598-3
3. Davari, S., Zarandi, M.H.F., Hemmati, A.: Maximal covering location problem (MCLP) with fuzzy travel times. Expert Syst. Appl. **38**(12), 14535–14541 (2011). https://doi.org/10.1016/j.eswa.2011.05.031
4. Lorena, L.A.N., Pereira, M.A.: A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillman's edition. Int. J. Ind. Eng. **9**(1), 57–67 (2002)

5. Zarandi, M.H.F., Davari, S., Sisakht, S.A.H.: The large scale maximal covering location problem. Sci. Iran. **18**(6), 1564–1570 (2011). https://doi.org/10.1016/j.scient.2011.11.008
6. Berman, O., Krass, D.: The generalized maximal covering location problem. Comput. Oper. Res. **29**(6), 563–581 (2002). https://doi.org/10.1016/S0305-0548(01)00079-X
7. Máximo, V.R., Nascimento, M.C.V., Carvalho, A.C.: Intelligent-guided adaptive search for the maximum covering location problem. Comput. Oper. Res. **78**, 129–137 (2017). https://doi.org/10.1016/j.cor.2016.08.018
8. Bagherinejad, J., Bashiri, M., Nikzad, H.: General form of a cooperative gradual maximal covering location problem. J. Ind. Eng. Int. **14**(2), 241–253 (2017). https://doi.org/10.1007/s40092-017-0219-5
9. Saifullah, C.M.K., Islam M.R.: Solving shortest common supersequence problem using chemical reaction optimization. In: 5th International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, pp. 50–55 (2016). https://doi.org/10.1109/ICIEV.2016.7760187
10. Kabir, R., Islam, R.: Chemical reaction optimization for RNA structure prediction. Appl. Intell. **49**(2), 352–375 (2018). https://doi.org/10.1007/s10489-018-1281-4
11. Mahmud M.R., Pritom, R.M., Islam, M.R.: Optimization of collaborative transportation scheduling in supply chain management with TPL using chemical reaction optimization. In: 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, pp. 1–6 (2017). https://doi.org/10.1109/ICCITECHN.2017.8281767
12. Islam, M.R., Islam, M.S., Sakeef, N.: RNA Secondary Structure Prediction with Pseudoknots using Chemical Reaction Optimization Algorithm. IEEE/ACM Trans. Comput. Biol. Bioinf. **18**(3), 1195–1207 (2021). https://doi.org/10.1109/TCBB.2019.2936570
13. Islam, M.R., Smrity, R.A., Chatterjee, S., Mahmud, M.R.: Optimization of protein folding using chemical reaction optimization in HP cubic lattice model. Neural Comput. Appl. **32**(8), 3117–3134 (2019). https://doi.org/10.1007/s00521-019-04447-8
14. Li, J.Q., Pan, Q.K.: Chemical-reaction optimization for flexible job-shop scheduling problems with maintenance activity. Appl. Soft Comput. **12**(9), 2896–2912 (2012). https://doi.org/10.1016/j.asoc.2012.04.012
15. Bargaoui, H., Driss, O.B., Ghédira, K.: A novel chemical reaction optimization for the distributed permutation flowshop scheduling problem with makespan criterion. Comput. Ind. Eng. **111**, 239–250 (2017). https://doi.org/10.1016/j.cie.2017.07.020
16. Zain, A.M., Yousif, A.: Chemical reaction optimization (CRO) for cloud job scheduling. SN Appl. Sci. **2**(1), 1–12 (2019). https://doi.org/10.1007/s42452-019-1758-8
17. Megiddo, N., Zemel, E., Hakim, S.L.: The maximum coverage location problem. SIAM J. Algebraic Discret. Methods **4**(2), 253–261 (1983). https://doi.org/10.1137/0604028
18. Lam, A.Y.S., Li, V.O.K.: Chemical reaction optimization: a tutorial. Math. Comput. Model. **4**(1), 3–17 (2012). https://doi.org/10.1007/s12293-012-0075-1