

# A hybrid artificial bee colony algorithm for the cooperative maximum covering location problem

B. Jayalakshmi<sup>1</sup> · Alok Singh<sup>1</sup>

Received: 27 July 2015 / Accepted: 25 November 2015 / Published online: 7 December 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** This paper proposes a hybrid artificial bee colony algorithm for the cooperative maximum covering location problem (CMCLP) on a network. In location covering problems, it is assumed that each facility generates a signal whose strength decreases with the increase in distance and a demand point is considered to be covered if the total signal strength received by it from various facilities exceeds a certain threshold. The objective of the CMCLP is to locate the facilities in such a way that maximizes the total demands covered. The proposed hybrid approach obtained better quality solutions in comparison to the methods available in the literature.

**Keywords** Artificial bee colony algorithm · Cooperative maximum covering location problem · Heuristic · Networks · Swarm intelligence.

## 1 Introduction

The cooperative maximum covering location problem (CMCLP) was introduced by [3]. In this problem, it is assumed that each facility generates a signal whose strength decreases over distance, according to some signal strength function. A demand point receives the signals from all the facilities and it is considered as covered, if the strength of the combined signal received by it from all the

facilities exceeds a certain threshold. Hence, the CMCLP follows a cooperative coverage model, where all the facilities cooperate in providing the coverage. This coverage model differs from the one used in classical maximum covering location problem (MCLP) introduced by [5], where the signal received from the single closest facility determines whether a demand point is covered or not, thus it is referred as individual coverage model.

The individual coverage model may not always be appropriate as it may result in poor quality of solutions by underestimating the coverage provided by the group of facilities together. This will obviously result in deployment of more number of resources than actually needed. In fact, it was shown in [3], that the solution resulting from considering individual coverage only can be around twice as worse as the solution resulting from considering cooperative coverage.

Signal can be physical or non-physical. Physical signal such as radio signal propagates along the straight line path between two points, whereas a non-physical signal may not. An example of a non-physical signal is the service time of an ambulance to reach a particular point from its rest location. Rather than traveling on a straight line path, an ambulance has to make use of the road network to reach the desired point from its rest position.

The potential application for cooperative cover problems include situations, where the facilities are generally unreliable and the customer needs to be covered by multiple facilities in order to ensure a satisfactory coverage. Another application of cooperative cover includes models assuming partial coverage by the facilities, i.e., a single facility is not sufficient to provide the required coverage to any customer.

Application of both signal types, physical and non-physical, of the cooperative coverage models have been

---

✉ Alok Singh  
alokcs@uohyd.ernet.in

B. Jayalakshmi  
bjayalakshmi@uohyd.ac.in

<sup>1</sup> School of Computer and Information Sciences, University of Hyderabad, Hyderabad 500 046, India

discussed by [3]. For example, application such as, installation of light towers to provide adequate lighting to an area, location of warning sirens and location of cell phone towers belong to physical signal class. The main aspect of the physical signal class of cooperative cover models is—strength of the signal received by a particular demand point is the combined signal strength from all the facilities. In case of installation of light towers, a demand point is covered if the luminous intensity at that point exceeds a certain threshold value. Similarly, in case of installation of warning sirens, a demand point is covered if the intensity of the sound at that point exceeds a certain threshold. In case of location of cell towers, the demand point (mobile phone user) is considered as covered if the probability of establishing a successful connection exceeds a certain threshold value. On the other hand, applications such as emergency response system design, belong to non-physical signal class. In such systems, having a single facility within the coverage of demand point is not sufficient, to ensure the coverage all the time, because it may become busy at some point. To deal with these types of issues, the demand point must be within the coverage radius of several facilities.

Averbakh et al. [1] provided a practical example for cooperative coverage model on a network in the context of “Pizza Delivery System”. Consider a set of pizza centres promoting their service as “Delivery in 30 Minutes Only”. Suppose the pizza preparation time is distributed as a normal variable with the mean of 20 min and standard deviation of 2 min. Also suppose that the trading area is defined as the set of customers who can be served at least 75 % of time within 30 min. For an individual coverage model, the customers should be within the coverage of  $30 - 20 - 0.7 \times 2 = 8.6$  min ( $0.7 = 75\%$  of the standard normal distribution) to get the delivery within 30 min. For instance, a customer located at a distance of 15 min from the pizza centre has only 50 % chances to get the pizza delivery within 30 min time. Suppose this customer has another pizza center which is at a distance of 9 min away. The customer has only 69 % chance to get a pizza delivered within 30 min time from this alternate pizza center. So none of the centers cover this customers who is located outside the coverage areas of both centers. Assuming the existence of a dispatcher which has the ability to redirect the order to the center which is least busy at the moment and on-site preparation times at both the centers are independent of each other. Here in this particular situation, it is possible to deliver the order to the customer within 30 min time with the probability of  $1 - (1 - 0.5)(1 - 0.69) = 0.84$ , which meets the coverage standard. Hence, this example, clearly shows that the customer has better chances of being covered by assuming cooperative coverage rather than individual coverage. From this example,

we can also conclude that if we need to determine the locations of some fixed number of pizza centres so as to cover as large an area as possible with required level of coverage, the area covered with cooperative coverage model will always be greater than or equal to the area covered by the individual coverage model. Likewise, location of emergency services such as fire stations and ambulances can be determined using cooperative coverage model on a network in a more efficient manner than using individual coverage model.

In the literature, covering problems are analyzed in plane by [6, 20] and [7] as well as for network topology by [16] and [2] assuming the discrete demand. Covering problems are also analyzed under the assumption of continuously distributed demand by [19] and [9]. The planar version of the CMCLP is solved using both optimal and heuristic algorithms by [3]. Berman et al. [3] proposed an algorithm which is based on the big triangle and small triangle approach [8] for the *2-facility* problem with the assumption that one of the facilities location is known. For more than *2-facilities* problem, they proposed a heuristic method. The discrete version of the CMCLP was considered by [4], where the location space is restricted to the nodes of the network. The discrete version allowing the facilities to be located at both the nodes and along the edges is solved by [1], where they proposed greedy methods and local search based heuristics to solve the *p-facility* problem.

In this paper, we have proposed an artificial bee colony algorithm based approach for solving the CMCLP for network version of *p-facility* problem where facilities can be located both at the nodes and along the edges. The artificial bee colony (ABC) algorithm is a recently developed population based meta-heuristic, proposed by [10]. ABC algorithm is inspired by foraging behaviour of real honey bee swarms. ABC algorithm has been successfully applied to solve numerous optimization problems in various domains. Motivated by the success of ABC algorithm in solving different optimization problems in various domains, we have designed an ABC algorithm to solve the CMCLP. We have compared the results obtained through our approach with two interchange heuristic methods proposed in [1]. In comparison to these methods, our approach obtained better quality results on most of the instances.

The remaining part of this paper is structured as follows: Sect. 2 defines the CMCLP formally. In Sect. 3, we provide a brief introduction to ABC algorithm. Section 4 presents our ABC approach for the CMCLP. Section 5 reports the computational results and compares our approach with other approaches available in the literature. Finally, Sect. 6 presents some concluding remarks and directions for future research.

**Table 1** Summary of notations

$n$	The number of nodes or demand points, i.e., $n =  V $
$w_i$	The weight associated with demand point $i \in V$
$l_k$	The length associated with each edge $e_k \in E$
$X$	The location space
$p$	The number of facilities to locate
$d_i(x_j)$	The distance between demand point $i \in N$ and facility $j$
$\phi(d)$	The strength of the signal at distance $d$ from the facility, $\phi(d) = \max\{0, 1 - d/U\}$
$\Phi_i(x)$	The overall signal at point $i \in N$
$U$	The fraction of diameter of the network
$T$	The threshold for coverage

### 2 Formal problem definition

The CMCLP can be formally defined as follows: Let  $G = (V, E)$  be an undirected network, with  $V = (1, \dots, n)$  be the set of demand points and  $E = (e_1, \dots, e_n)$  be the set of edges connecting various demand points. Each demand point  $i$  has a non-negative real weight  $w_i$  indicating the total demand at this point. Each edge  $e_k$  has a positive length  $l_k$ .  $p$  facilities need to be located on the edges (including end demand points) to cover these demand points. A demand point is said to be covered, if the combined strength of the signal received by it from all the facilities is not less than a threshold  $T$ . The CMCLP seeks a location vector  $X_p$  for these  $p$  facilities so as to maximize the sumtotal of the weights of the covered demand points.

$$f(X_p, T) = \sum_{i: \Phi_i(X_p) \geq T} w_i$$

The signal strength function is defined as the sum of all signals received by a demand point  $i \in V$  from  $p$  facilities.

$$\Phi_i(X_p) = \sum_{k=1}^p \phi(d_i(x_k))$$

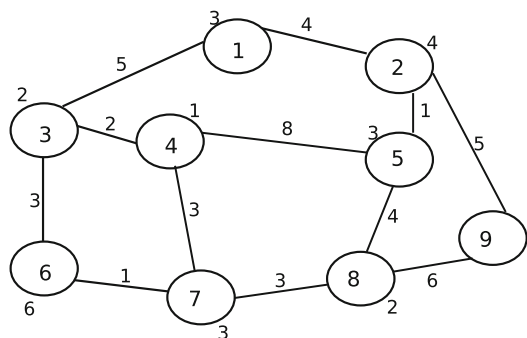
Location  $x$  of a facility along an edge  $e_k$  joining demand point  $j_1$  with  $j_2$  ( $e_k = (j_1, j_2)$ ) is represented by an ordered pair  $(l_k, t)$  where  $t$  is the relative distance of  $x$  from  $j_1$  with respect to  $l_k$ , i.e.,  $0 \leq t \leq 1$ . The distance from a node  $i \in V$  to this facility is defined as follows

$$d(i, x) = \min\{d(i, j_1) + t \times l_k, d(i, j_2) + (1 - t) \times l_k\}$$

where  $d(i, j_1)$  and  $d(i, j_2)$  are the lengths of the shortest paths connecting node  $i$  to node  $j_1$  and node  $j_2$  respectively. The notations used above are summarized in the Table 1.

To illustrate CMCLP, consider the network depicted in Fig. 1. Let  $p = 3$ ,  $T = 0.5$ , and the signal strength function be defined as  $\phi(d) = \max\{0, 1 - \frac{d}{10}\}$ .

One possible solution that we can obtain by locating the facilities along the edges as well as on nodes is  $X = \{(4), ([7, 8], 0.6), ([1, 2], 0.4)\}$ , where the first point  $p_1$  is located on node 4, the second point  $p_2$  is located on edge  $[7, 8]$  at a relative distance of  $t = 0.6$  from node 7, and the



**Fig. 1** Illustrating CMCLP

last point  $p_3$  is located on edge  $[1, 2]$  at a relative distance of  $t = 0.4$  from node 1. This solution produces an objective value of 29 covering all nodes. In this example, we have selected the points randomly. There might be a possibility of some other solutions existing for this network.

### 3 Overview of ABC algorithm

The artificial bee colony (ABC) algorithm proposed by Karaboga in 2005 is a population based meta-heuristic algorithm, which is inspired from the intelligent behavior of the foraging honey bees [10]. In a bee colony, there are three types of bees: employed, onlooker and scout. Employed bees are those bees which are currently exploiting a food source. The responsibility of the employee bees is to bring loads of nectar to the hive and share the information about their food sources with other bees waiting in the hive. The waiting bees are the onlookers. The onlookers then choose a food source with a probability directly proportional to its quality and becomes employed. Scout bees search for new food sources in the vicinity of the hive and they become employed as soon as they find a new food source. An employed bee whose food source becomes empty will abandons that food source and becomes either a scout or an onlooker.

Motivated by the foraging bees’ behavior described above, Karaboga developed ABC algorithm. This

algorithm was originally developed for solving optimization problems in continuous domain only, later, it has been extended to solve discrete optimization problems also [10–13, 15, 17, 18]. For a recent survey on ABC algorithm and its applications, interested readers may refer to [14].

In ABC algorithm also there are three groups of bees, viz. employed, onlooker and scout with functions similar to their real counterparts. In ABC algorithm, the food sources represent the possible solutions to the problem under consideration and the quality of a food source represents the fitness of the represented solution. The employed bees are associated with food sources. Always, there is a one-to-one association between food sources and employed bees, which means, the number of food sources is equal to the number of employee bees. Usually, but not always, the number of onlooker bees is also taken to be equal to number of employed bees. The ABC algorithm is a iterative search process, which starts with initializing the employee bees, with randomly generated food sources (solutions), then repeats through the cycles of the employed bee and onlooker bee phases.

In the employed bee phase, each employed bee generates a food source in the proximity of its associated food source and evaluates its quality. The method of determining a new food source in the proximity of a particular food source depends on the problem under consideration. If the quality of the new food source is better than the current one then the employed bee moves to the new food source leaving the old one. Otherwise, it remains at the old food source. When all the employed bees finish this process, then employed bee phase ends and onlooker bee phase begins.

Onlooker bee phase starts with sharing of information by employed bees about their food sources with the onlookers. Onlookers select the food sources depending on their quality, i.e., higher the value of the fitness of the solution represented by a food source, higher will be the chances of its selection. As a result of such a selection, good quality food sources will get more chance for selection by the onlookers. After all onlookers select the food sources, they determine the food sources in the proximity of their selected food sources in a manner similar to the employed bees and evaluate their fitness. Among all the neighboring food sources generated by the onlookers who chose food source  $i$  and the food source  $i$  itself, the best quality food source is determined. This best food source will be updated as food source  $i$  for next iteration. The onlooker bee phase ends once all food sources are updated, and then the next iteration of the ABC algorithm begins. The algorithm is repeated until the termination condition is satisfied. If a solution associated with any employed bee does not improve over some specific number of iterations, then that food source is considered as exhausted and it is discarded by its associated employee bee and that employee bee becomes scout. Such scouts are converted

back into employed bees by associating them with newly generated solutions. Usually, these new solutions are generated randomly in the same manner as initial employed bee solutions.

In the employed bee phase, every solution is given a fair chance to improve itself, whereas in the onlooker bee phase, because of the selection policy used by the onlookers as mentioned above good quality solutions get more chance to improve themselves in comparison to poor quality solutions. This inclination towards selecting good quality solutions may produce better quality solutions, as there will be higher chances of finding even better solutions within the proximity of good solutions. However, if a solution is locally optimal, then no better solution exists in its proximity and any attempt to improve it will always fail. The concept of scout bees helps in this situation. If a solution is not improved over certain number of iterations then it is assumed to be locally optimal and is discarded by making its associated employed bee a scout. A new solution is generated for this scout bee to make it employed again. This new solution is created either in the same manner as an initial solution or by perturbing an existing solution. Hence by utilizing the concept of scout bees, solutions which are not improved since long are replaced with new solutions. In a robust search process, the balance between the exploration and exploitation must be maintained. In the ABC algorithm, employed bees and onlooker bees carry out the exploitation, and scouts perform the exploration. A proper balance need to be maintained between exploration and exploitation by appropriately setting the number of iterations without improvement in the ABC algorithm after which an employed bee abandons a solution and becomes scout.

## 4 ABC approach for the CMCLP

This section presents salient features of our ABC approach for the CMCLP.

### 4.1 Fitness of a solution

We follow the same two level approach as used in [1] for determining the fitness of a solution. We say that a solution  $X'$  is better than another solution  $X$ , if solution  $X'$  either has a larger objective function value,

$$f(X', T) > f(X, T)$$

or, if solution  $X$  and solution  $X'$  have same objective function value, but solution  $X'$  provides larger total coverage.

$$\sum_{i \in V} \Phi_i(X') > \sum_{i \in V} \Phi_i(X)$$

### 4.2 Method for selecting a food source for an onlooker

We have used the binary tournament selection method for selecting a food source for an onlooker. In the binary tournament selection method, two food sources are selected uniformly at random, and their fitness is compared. The better of the two food sources is selected with the probability  $p_{onl}$ . Otherwise, the worse of the two food sources is selected, i.e., the probability of selection of the worst solution is  $1 - p_{onl}$ . The Pseudo code for the binary tournament selection method is as follows:

---

**Algorithm 1:** Binary tournament selection

---

```

Select two solutions  $e_1$  &  $e_2$  randomly from the employee bee solutions;
 $p :=$  generate a random number between 0 and 1;
if  $p \leq p_{onl}$  then
     $\perp$  return the best among  $e_1$  and  $e_2$ 
else
     $\perp$  return the worst among  $e_1$  and  $e_2$ 
    
```

---

### 4.3 Initial solution

To generate an initial solution, we have followed a method which is a mix of greediness and randomness. In this method one facility at a time will be added to the current partial solution starting with an empty solution. In each iteration, to add a facility to the current partial solution  $S$ , we compute the set  $Y$  of all points  $x \in G$  (including nodes  $\in V$ ) that provide exact or greater coverage for yet uncovered nodes  $V'$ . The  $x$  points are considered on edges at equal intervals separated by a relative distance of 0.1. The condition of exact or greater coverage is based on the signal strength function defined. With the choice of the signal strength function used and/or the way we are considering the points on the edges, it may happen during the iterations of the greedy method that there may not exist even one point which can provide exact coverage (specially in case of large instances). So we have chosen points providing either exact or greater coverage. Next, we evaluate the points based on how much coverage they are providing, then choose  $R$  best points from the set  $Y$ , and select one point randomly from  $R$ .

$$Y = V \cup \{x \in G | \phi_i(x) \geq T_i(X)\} \text{ for some } i \in V'$$

After locating a facility, the threshold is updated as follows:

$$T_i(S) = \max\{0, T - \Phi_i((S))\} \text{ for } i \in V$$

This process is repeated till the desired number of facilities are located.

### 4.4 Neighboring solution generation

To generate a solution  $X'$  in the neighborhood of solution  $X$ , we delete  $F$  facilities from  $X$  randomly. Then,  $F$  facilities are added one-by-one from the set  $F^{new}$  containing all points  $x \in G$  (including nodes  $\in V$ ) which provide exact coverage for at least one yet un covered node in  $V'$ . A formal definition of  $F^{new}$  is given below.

$$F^{new} = Z - X$$

where

$$Z = V \cup \{x \in G | \phi_i(x) = T_i(X)\} \text{ for some } i \in V'$$

To add each facility, we first choose  $R$  best points from  $F^{new}$  and then choose one randomly from  $R$ . Once a facility has been added,  $F^{new}$  is updated. The manner in which facilities are added is similar to Greedy 1 heuristic of [1] except for the fact that Greedy 1 always choose the best point, whereas we choose randomly a point from among  $R$  best points.

### 4.5 Other features

We have used different number of employee bees and onlooker bees. If an employed bee solution does not improve for a specified number of iterations  $limit$  then the associated employed bee becomes scout. There is no restriction on the number of scout bees in an iteration. The number of scouts in a particular iteration depends on the number of employed bee solutions which have not improved for exactly  $limit$  number of iterations. The scout bee is again made employed by assigning it to a new solution which is generated in the same manner as one of our initial solution.

### 4.6 Local search

We have used a local search on the best solution obtained through ABC algorithm in a bid to improve it further. In this local search, each facility  $x$  in a solution  $S$  is considered one-by-one and the best point  $b_x$  to relocate it is determined. For this, we compute the set  $F^{new}$  for solution  $(S \setminus \{x\})$  and find the best point  $b_x$  to locate the next facility. If the solution  $(S \setminus \{x\}) \cup \{b_x\}$  is better than  $S$  then we replace  $S$  with this new solution. This process is repeated till each facility is considered once.

Algorithm 2 provides the pseudo-code for our hybrid ABC approach, where  $n_e$  and  $n_o$  are respectively the number of employed bees and number of onlooker bees.  $Generate\_Neighbor(e_i)$  and  $Binary\_Tournament(e_1, e_2, \dots, e_n)$  are two functions. The former function generates a solution in the neighborhood of solution  $e_i$  as per Sect. 4.4, whereas the

latter selects a solution for an onlooker bee from all employed bee solutions  $e_1, e_2, \dots, e_n$  as per Sect. 4.2.

---

**Algorithm 2:** Pseudo-Code of our Hybrid ABC Algorithm
 

---

```

Randomly generate  $n_e$  solutions  $e_1, e_2, \dots, e_{n_e}$ ;
best_sol := best solution among  $e_1, e_2, \dots, e_{n_e}$ ;
while termination condition is not satisfied do
  for  $i = 1$  to  $n_e$  do
     $e'_i := \text{Generate\_Neighbor}(e_i)$ ;
    if  $e'_i$  is better than  $e_i$  then
       $e_i := e'_i$ 
    if  $e'_i$  is better than best_sol then
      best_sol :=  $e'_i$ 
  for  $i = 1$  to  $n_o$  do
     $k_i := \text{Binary\_Tournament}(e_1, e_2, \dots, e_n)$ ;
     $onl_i := \text{Generate\_Neighbor}(e_{k_i})$ ;
    if  $onl_i$  is better than best_sol then
      best_sol :=  $onl_i$ 
  for  $i = 1$  to  $n_o$  do
    if  $onl_i$  is better than  $e_{k_i}$  then
       $e_{k_i} := onl_i$ 
  for  $i = 1$  to  $n_e$  do
    if  $e_i$  has not improved over limit iterations then
      replace  $e_i$  with a random solution;
best_sol := Local_search(best_sol);
return best_sol;

```

---

## 5 Computational results

To test our ABC approach, we have used the same instances as used in [1]. These instances are generated randomly in the following manner: For each combination of  $(n, dgr)$  where  $n$  is the number of nodes and  $dgr$  is the average degree of nodes, five instances were generated by [1]. In these instances  $n \in \{40, 60, 80, 100, 120, 140, 160, 180, 200\}$  and  $dgr \in \{5, 6, 7\}$ . Like [1], we have done all computational experiments by taking  $p = 3, 4, 5$  for  $n = 40, 60, 80$ ;  $p = 4, 5, 6$  for  $n = 100, 120, 140$ ; and  $p = 5, 6, 7$  for  $n = 160, 180, 200$ . Three different values have been used for signal threshold values  $T$ , i.e.,  $T \in \{0.6, 0.8, 1.0\}$  and linear signal strength function  $\phi(d) = \max\{0, 1 - d/U\}$  has been used like in [1]. The parameter  $U$  was determined in [1] as a fraction of the diameter of the network and  $U_{\%} \in \{0.15, 0.25, 0.35\}$  for  $T \in \{0.6, 0.8\}$  and  $U_{\%} \in \{0.2, 0.3, 0.4\}$  for  $T = 1.0$  were used. Overall, there are 3625 instances.

Our ABC approach has been implemented in C and executed on a Linux based Intel Core i5 2400 system with 4 GB memory running at 3.10 GHz. In all our computational experiments, the number of employed bees ( $n_e$ ) is taken to be 10, the number of onlooker bees ( $n_o$ ) is taken to be 20,  $limit$  is set to 50,  $p_{onl}$  is set to 0.9 for  $U_{\%} = 0.15, 0.25, 0.35$  and  $T = 0.6$ , it is set to 0.8 for remaining combinations of  $U$  and  $T$ , and  $R = 20$ . We have taken  $F =$

**Table 2** Comparison between our ABC approach and I1 and I2 heuristics of [1]

$T$	$U_{\%}$	%(ABC, I1)	%(ABC, I2)	R(I1)	R(I2)	R(ABC)
0.6	0.15	1.28	1.21	42.36	25.28	351.62
0.6	0.25	12.84	12.76	99.99	67.21	253.26
0.6	0.35	1.1	1.07	80.61	104.49	244.21
0.8	0.15	4.88	4.71	39.07	28.93	366.45
0.8	0.25	22.36	21.98	84.42	78.00	271.49
0.8	0.35	5.34	5.06	118.80	141.13	257.82
1.0	0.2	35.98	35.52	36.26	29.48	263.43
1.0	0.3	25.15	24.86	122.42	108.10	278.56
1.0	0.4	12.08	12.09	82.56	153.21	206.29

2 for  $n = 40, 60, 80$  and  $p = 3, 4, 5$ ,  $F = 3$  for  $n = 100, 120, 140$  and  $p = 4, 5, 6$ ,  $F = 4$  for  $n = 160, 180, 200$  and  $p = 5, 6, 7$ . Our ABC approach terminates after 500 iterations. All these parameter values were chosen empirically after a large number of trials.

We have compared the results of our ABC approach with interchange heuristics I1 and I2 proposed in [1]. Results are reported in the same format as used in [1]. Averbakh et al. [1] reported the average percentage improvement in solution quality by I2 with respect to I1 over all the instances with same  $T$  and  $U_{\%}$ . The percentage improvement in solution quality by method A over method B on a particular instance is  $100 \times \frac{\Phi_i(S_A) - \Phi_i(S_B)}{\Phi_i(S_B)}$  where  $S_A$  and  $S_B$  are the solutions obtained by methods A and B respectively on the particular instance under consideration. However, average execution times are reported over all instances with  $n = 200$  only, i.e., all instances having maximum number of nodes. Table 2 reports the results of I1, I2 and our ABC approach. Data for I1 and I2 have been obtained from the corresponding author of [1] through personal communication. The first column in Table 2 represents the thresholds ( $T$  value), second column represents  $U_{\%}$  values, third column (%(ABC,I1)) reports the average percentage improvement of our ABC approach over I1, fourth column (%(ABC,I2)) reports the average percentage improvement of our ABC approach over I2. And the last 3 columns (R(.)) report the average execution times of various approaches on the largest instance ( $n = 200$ ). There are 405 instances with same  $T$  and  $U_{\%}$  and hence, reported average percentage improvements are averaged over these 405 instances. Similarly, reported execution times are averaged over 45 instances with 200 nodes.

The results shows that the performance of our ABC algorithm is better compared to the Interchange heuristics on all classes of instances with same  $T$  and  $U_{\%}$  values. In fact, average percentage improvement in solution quality

by our approach over I1 and I2 is always positive. Our approach on an average provide larger coverage than already existing methods. I1 and I2 were executed on a system having 3.31 GHz AMD Phenom II processor and 16 GB RAM which is different from the system used to execute our ABC approach. Therefore, execution times can not be compared directly. However, it is clear that our approach is slower than interchange heuristics I1 and I2.

## 6 Conclusions

In this paper, we have proposed an ABC algorithm based approach for the CMCLP and compared it with two interchange based heuristic methods proposed in the literature. Our ABC algorithm outperformed these two methods in terms of solution quality. However, its slower than these methods.

Our ABC algorithm based approach is the first metaheuristic approach for the CMCLP. Averbakh et al. [1] also experimented with tabu search and variable neighborhood search, but results obtained by these approaches were only slightly better than those obtained by interchange heuristics and hence they did not present these metaheuristics. Therefore, population based metaheuristics seem more appropriate for this problem. As a future work, we intend to investigate this further by developing some other metaheuristics for the CMCLP and compare them with our ABC approach and two heuristics of [1]. Approaches similar to our ABC approach can be developed for other related problems employing cooperative coverage model.

**Acknowledgments** Authors would like to place on record their sincere thanks to Dr. Jörg Kalcsics for providing the CMCLP test instances and their solution values obtained by various approaches proposed in [1]. Authors are also grateful to two anonymous reviewers for their valuable comments and suggestions which helped in improving the quality of this manuscript.

## References

1. Averbakh I, Berman O, Krass D, Kalcsics J, Nickel S (2014) Cooperative covering problems on networks. *Networks* 63(4):334–349
2. Berman O (1994) The p maximal cover—p partial center problem on networks. *Eur J Oper Res* 72(2):432–442
3. Berman O, Drezner Z, Krass D (2010) Cooperative cover location problems: the planar case. *IIE Trans* 42(3):232–246
4. Berman O, Drezner Z, Krass D (2011) Discrete cooperative covering problems. *J Oper Res Soc* 62(11):2002–2012
5. Church R, ReVelle C (1974) The maximal covering location problem. *Pap Reg Sci Assoc* 32(1):101–118
6. Drezner Z (1981) On a modified one-center model. *Manag Sci* 27(7):848–851
7. Drezner Z (1986) The p-cover problem. *Eur J Oper Res* 26(2):312–313
8. Drezner Z, Suzuki A (2004) The big triangle small triangle method for the solution of nonconvex facility location problems. *Oper Res* 52(1):128–135
9. Drezner Z, Suzuki A (2010) Covering continuous demand in the plane. *J Oper Res Soc* 61(5):878–881
10. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. In: Technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey
11. Karaboga D, Akay B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim* 39(3):459–471
12. Karaboga D, Akay B (2008) On the performance of artificial bee colony (ABC) algorithm. *Appl Soft Comput* 8(1):687–697
13. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. *Appl Math Comput* 214(1):108–132
14. Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif Intell Rev* 42(1):21–57
15. Pan QK, Tasgetiren MF, Suganthan P, Chua T (2011) A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Inf Sci* 181(12):2455–2468
16. ReVelle C (1986) The maximum capture or sphere of influence location problem: Hotelling revisited on a network. *J Reg Sci* 26(2):343–358
17. Singh A (2009) An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Appl Soft Comput* 9(2):625–631
18. Singh A, Sundar S (2011) An artificial bee colony algorithm for the minimum routing cost spanning tree problem. *Soft Comput* 15(12):2489–2499
19. Suzuki A, Drezner Z (1996) The p-center location problem in an area. *Locat Sci* 4(1–2):69–82
20. Watson-Gandy C (1982) Heuristic procedures for the m-partial cover problem on a plane. *Eur J Oper Res* 11(2):149–157