Improving Best Cost in Ant Colony Optimization with obstacles

Balraj Singh[1], Er. Harpreet Kaur[2]

[1]Research Scholar (M.Tech. CE), Punjabi University, Patiala, Punjab, India.

[2]Assistant Professor, University College of Engineering, Dept. of CE, Punjabi University, Patiala, Punjab, India.

Abstract – Ant Colony Optimization (ACO) algorithms are motivated by the searching behavior of ants in nature. The research paper focuses on the working principle of ACO with a special condition of obstacles been placed between the starting point (source) and the food source (destination). The optimization code has been implemented using MATLAB with three obstacles placed between source and destination. The implementation has been performed with three different cases with a varying number of iterations to calculate the Best Cost in each case.

Keywords – Ants, Ant Colony Optimization, Best Cost, Best Path, obstacles.

I. INTRODUCTION

Ant colony optimization (ACO) algorithms are inspired by the foraging behavior of ants in nature. Ant colony optimization (ACO) methodology is based on the ant's competence of finding the shortest path from the nest to a food source [1, 2]. An ant repeatedly hops from one location to another to ultimately reach the destination. In nature, some species of ants in searching for food will leave chemicals that can be smelled by others on the route, called pheromones. By releasing pheromones, ants can mark the route they have walked, providing clues for other ants foraging for food [3]. As time and the number of foraging ants increase, the concentration of pheromone in the environment will change, based on which ants can gradually find the shortest route between their nest and the food. This technique is developed by observing the food searching efforts of ant clusters [4, 5]. The ant colony follows an organized and smart technique for approaching the food source. The details of the technique are modeled with mathematical tools, and then the approach is transformed into an optimization problem framework to utilize for engineering problems such as the search area is defined as a graph and the agents (ants) are described as moving point on this graph [6, 7]. As the agents move on the graph, a simulation version of the pheromone secreting model is realized with a stochastic approach to mark the most popular paths through the source [8]. Each ant starts to move from randomly selected points on the graph. The connection line from the starting point to the target describes a path and each path is categorized with pheromone level and correlating heuristic value so that, higher of these parameters for a path gives rise to higher the probability an ant prefers this shorter path through the source. The rest of the ants use the pheromone deposited on the path for searching a more promising direction through the food target [9, 10]. Then, this iterative procedure goes on until each of the ants finishes their travel for the food and the pheromone level is updated on each path visited by the ants. Consequently, each ant provides a solution and, at least, one path among the solutions should fulfill the termination criterion to finish all procedures. As the main characteristic of this technique depends on the pheromone level on the route through the food source, the higher the depositing pheromone load, the higher the optimality a solution is categorized [11, 12].

II. IMPLEMENTATION OF ANT COLONY OPTIMIZATION

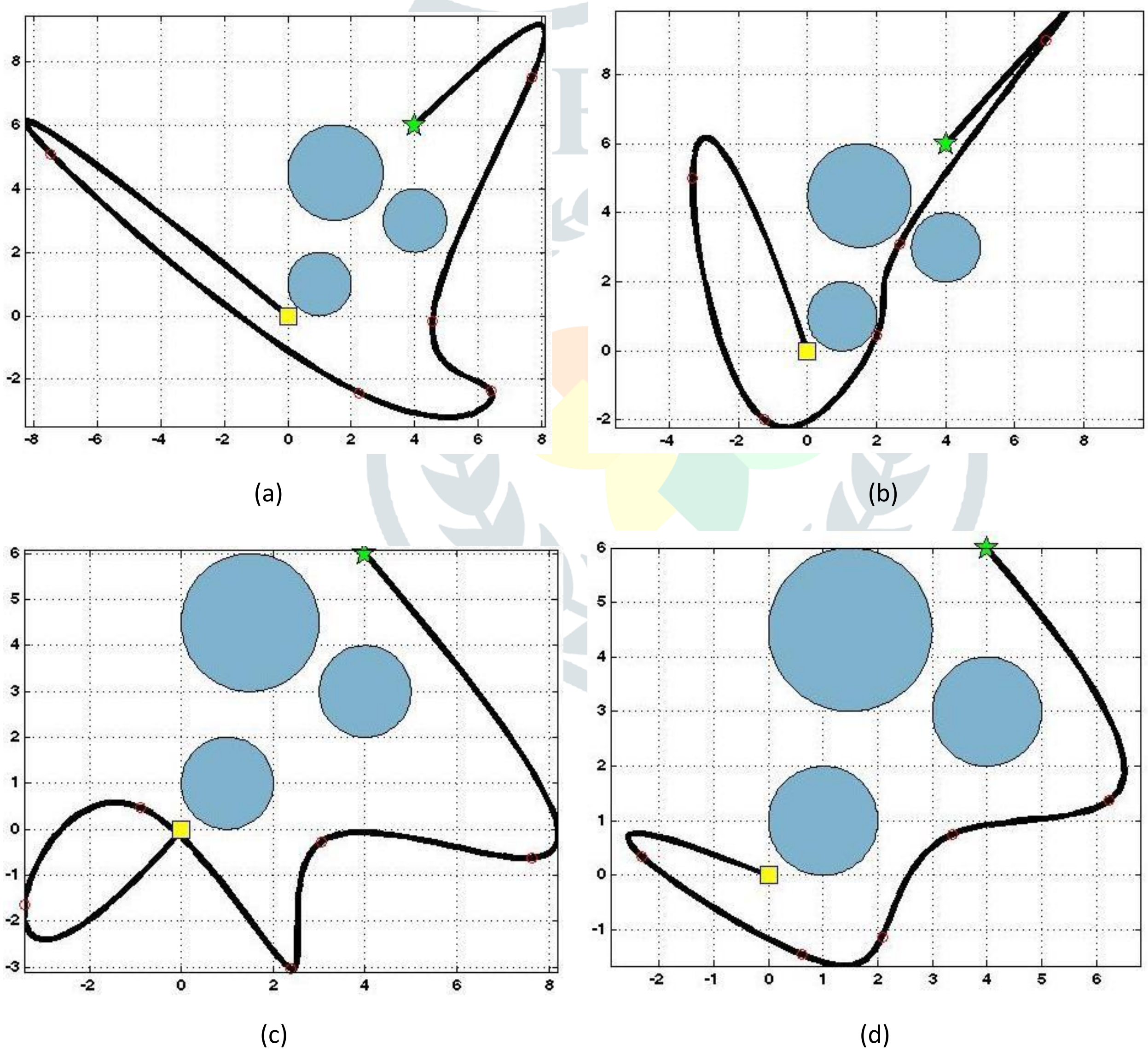
This section focuses on the implementation and working of ACO to find Best Path with Best Cost between the starting point (source) and food source (destination) with three obstacles placed on the way. The ants are supposed to reach the destination avoiding obstacles. Parameters include inertia weight which symbolizes the movement of the ants while adjusting their velocities and positions. Inertia Weight Damping Ratio determines the ratio between rates of ant's previous velocity to its velocity at the current time step. Personal learning and global learning coefficients are the two constants. The implementation has been conducted with three different cases having variation in the number of iterations.

Case 1:

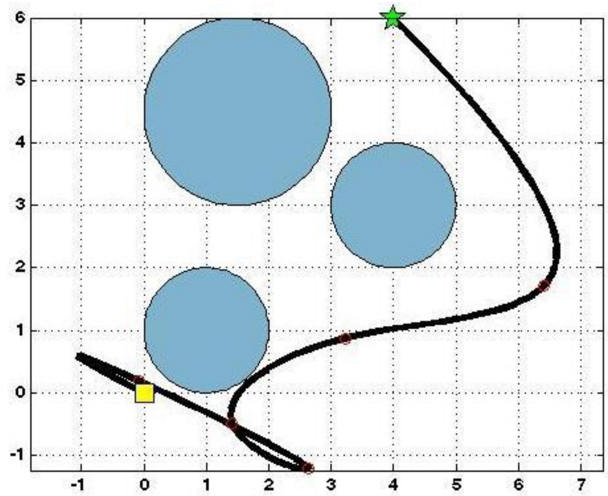
The readings of different parameters in case 1 are mentioned as under.

- Maximum Number of Iterations - 10
- Population Size (Swarm Size) - 2
- Inertia Weight - 1
- Inertia Weight Damping Ratio - .98
- Personal Learning Coefficient - 1.5
- Global Learning Coefficient - 1.5
- Number of handle points - 5

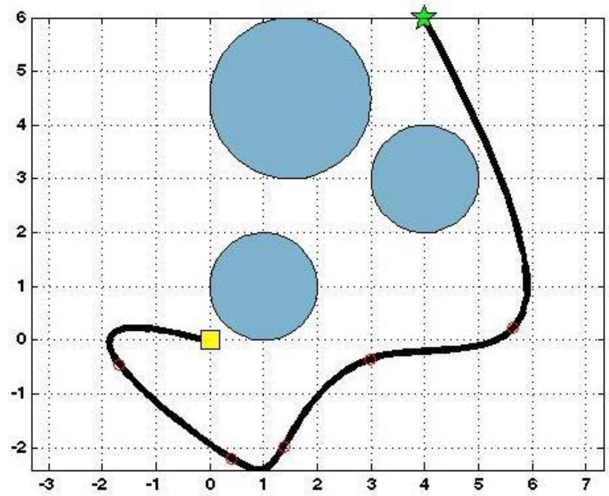
The implementation of ACO as per parameter values mentioned in Case 1 are shown in Fig. 1.



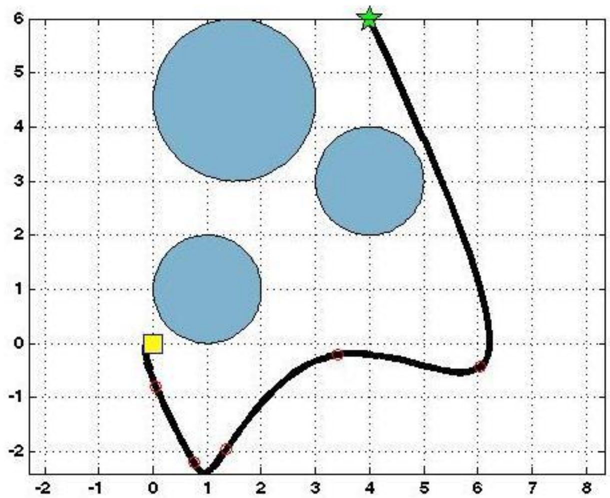
(a) (b) (c) (d)


6
5
4
3
2
1
0
-1
-1 0 1 2 3 4 5 6 7

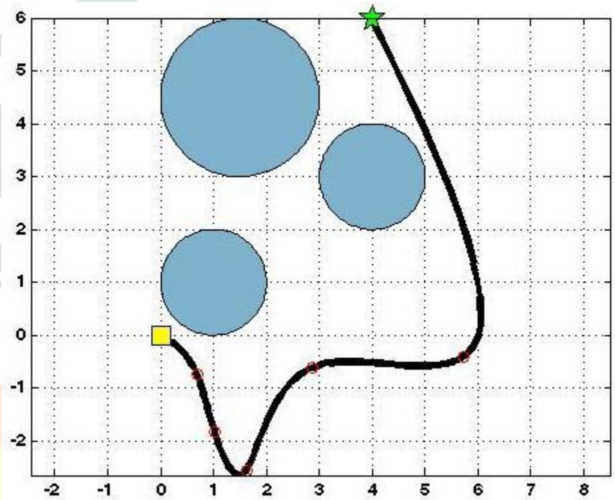
(e)


6
5
4
3
2
1
0
-1
-2
-3 -2 -1 0 1 2 3 4 5 6 7

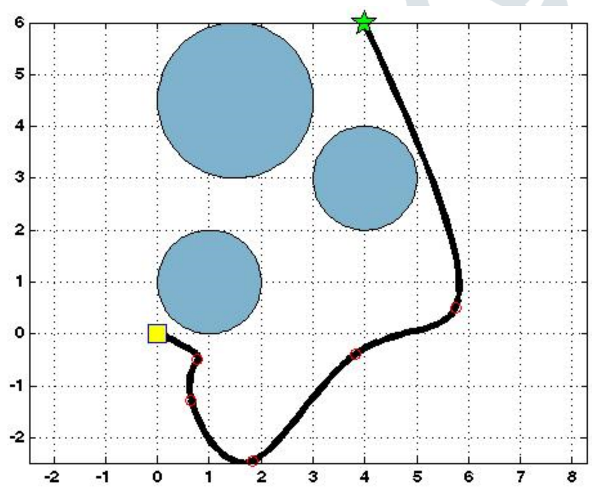
(f)


6
5
4
3
2
1
0
-1
-2
-2 -1 0 1 2 3 4 5 6 7 8

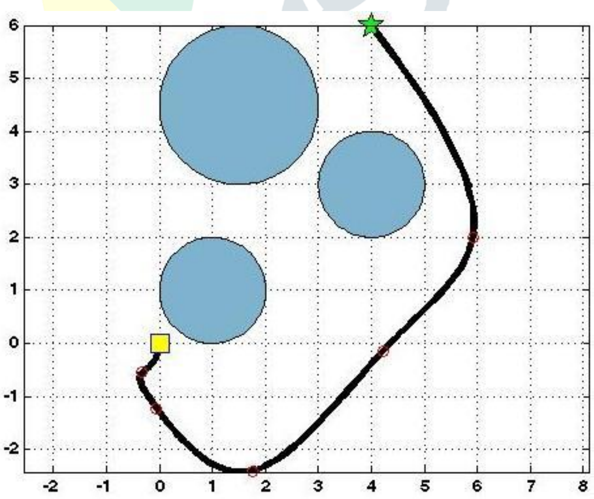
(g)


6
5
4
3
2
1
0
-1
-2
-2 -1 0 1 2 3 4 5 6 7 8

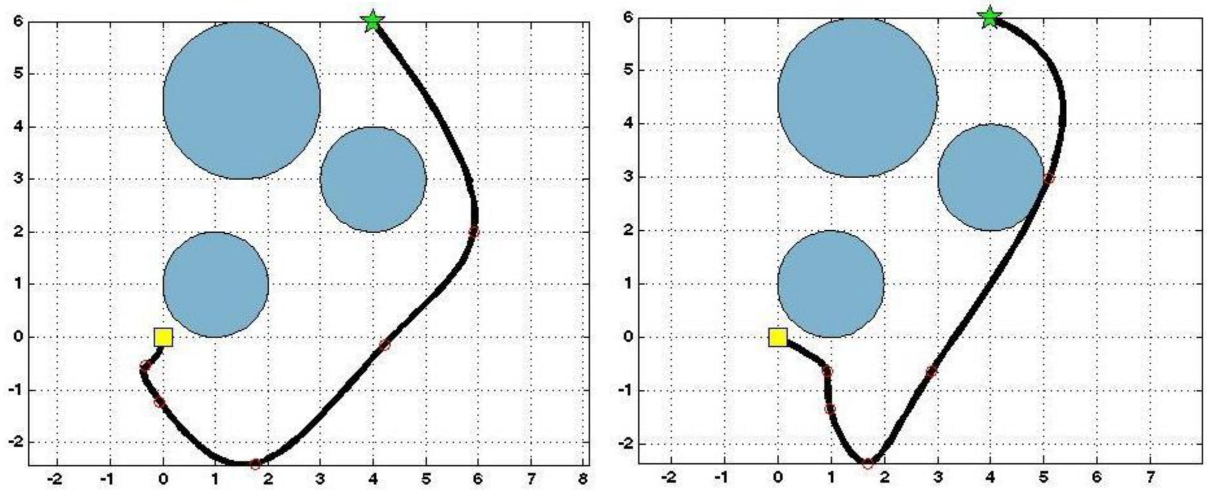
(h)


6
5
4
3
2
1
0
-1
-2
-2 -1 0 1 2 3 4 5 6 7 8

(i)


6
5
4
3
2
1
0
-1
-2
-2 -1 0 1 2 3 4 5 6 7 8

(j)



(k) (l)

Fig. 1. The figure shows different stages of ACO for calculating Best Cost involved in finding Best Path from source (yellow box) to the destination (green star) with three obstacles (blue circles) on the way

Fig. 2 depicts the readings of Best Cost at different iterations involved in reaching from source to destination.

```
Iteration 1: Best Cost = 47.0603 @
Iteration 2: Best Cost = 36.6231 @
Iteration 3: Best Cost = 29.5002 @
Iteration 4: Best Cost = 19.1815 @
Iteration 5: Best Cost = 19.1815 @
Iteration 6: Best Cost = 17.9201 @
Iteration 7: Best Cost = 17.7978 @
Iteration 8: Best Cost = 17.7978 @
Iteration 9: Best Cost = 15.8466 @
Iteration 10: Best Cost = 15.8466 @
Iteration 11: Best Cost = 15.8466 @
Iteration 12: Best Cost = 15.8466 @
Iteration 13: Best Cost = 15.8466 @
Iteration 14: Best Cost = 15.8466 @
Iteration 15: Best Cost = 15.513 @
Iteration 16: Best Cost = 15.513 @
Iteration 17: Best Cost = 14.5335 @
Iteration 18: Best Cost = 14.4183 @
Iteration 19: Best Cost = 14.4183 @
Iteration 20: Best Cost = 13.1898 @
```

Fig. 2. The figure shows the readings of Best Cost at different iterations involved in reaching from source to destination

Fig. 3 shows the plotted graph based on obtained readings of Fig. 2. The X-axis represents the number of iterations and Y-axis represents the obtained Best Cost.

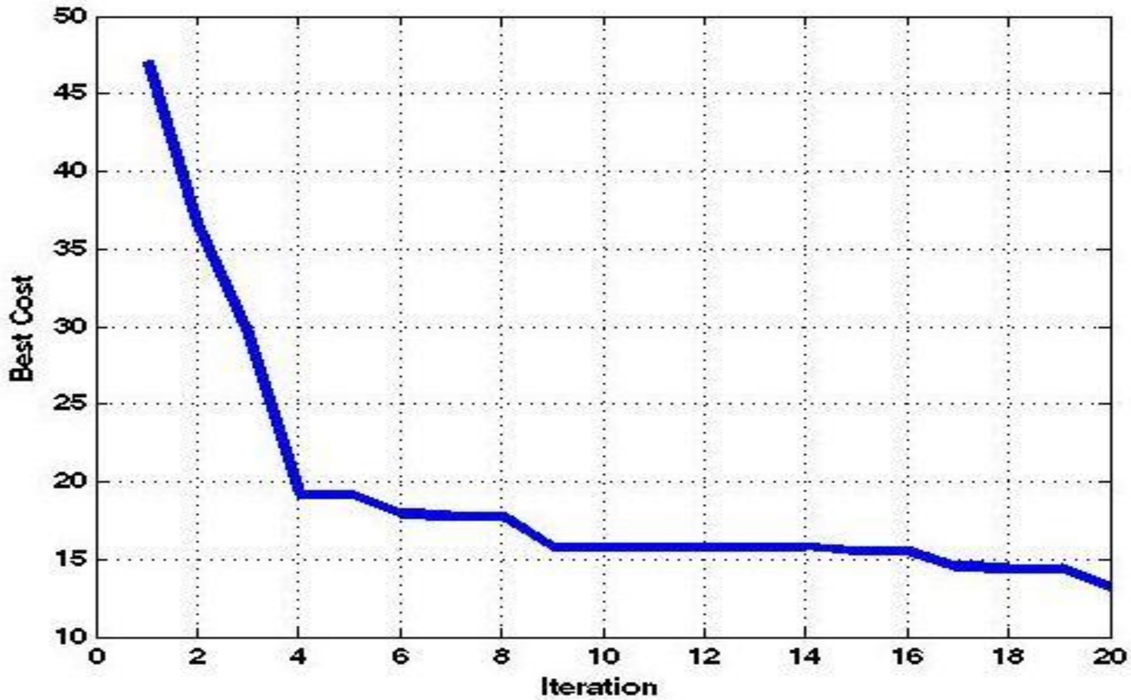


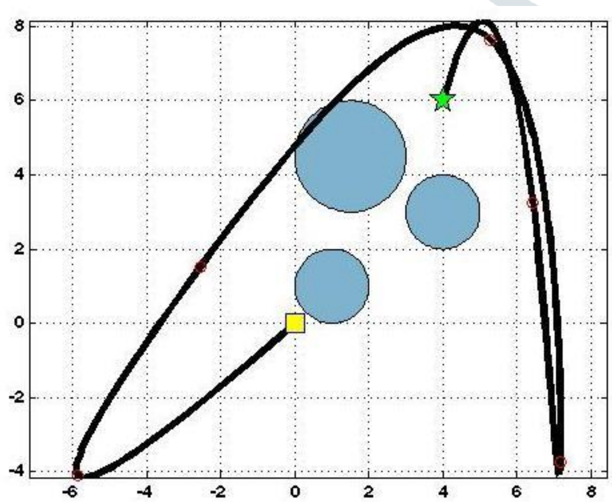
Fig. 3. The figures show the plotted graph based on Best Cost readings as per Case 1

Case 2:

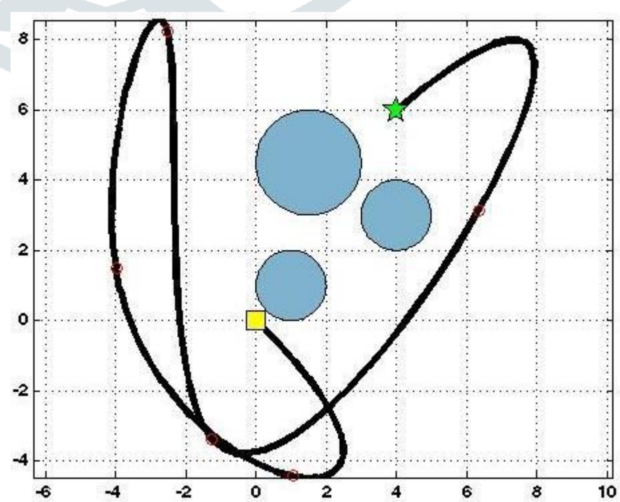
The readings of different parameters in case 2 are mentioned as under.

- Maximum Number of Iterations - 40
- Population Size (Swarm Size) - 2
- Inertia Weight - 1
- Inertia Weight Damping Ratio - .98
- Personal Learning Coefficient - 1.5
- Global Learning Coefficient - 1.5
- Number of handle points - 5

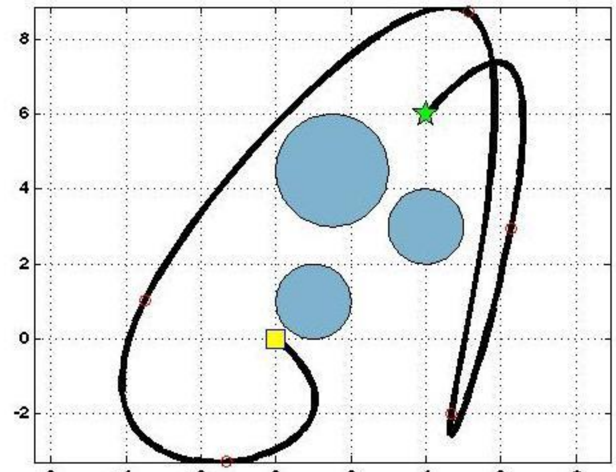
The implementation of ACO as per parameter values mentioned in Case 2 are shown in Fig. 4.



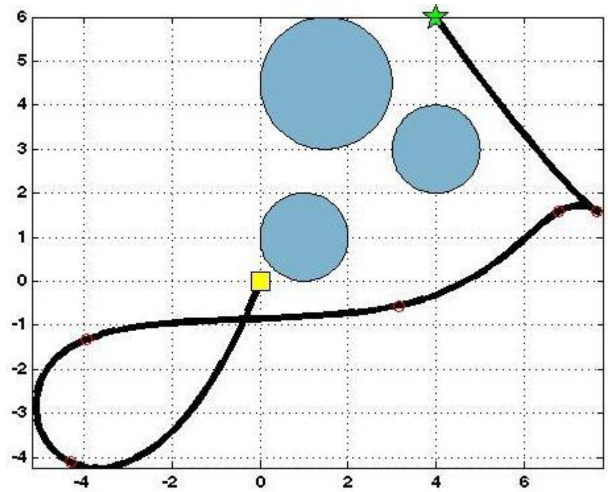
(a)



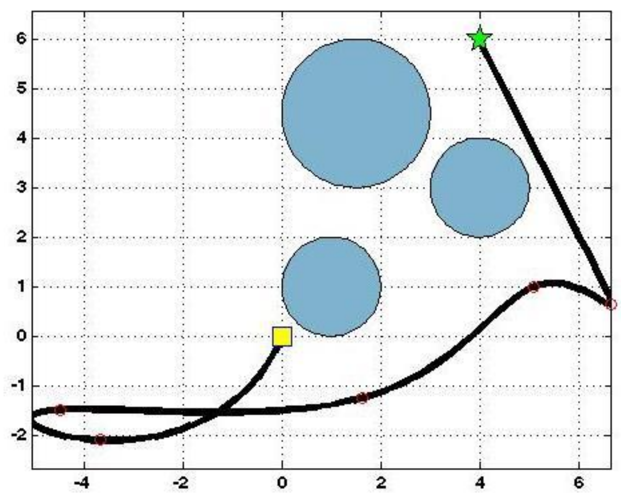
(b)



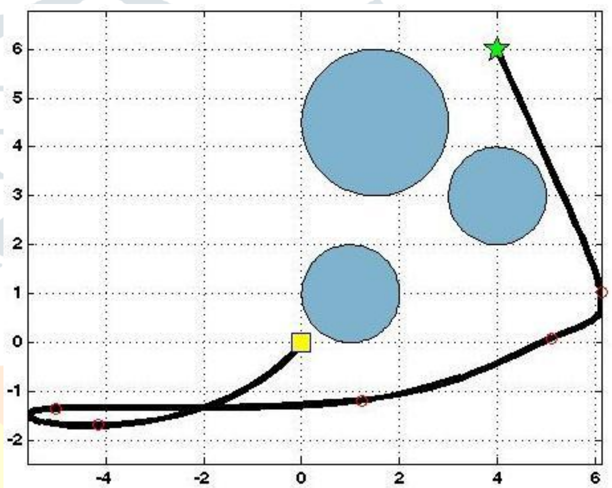
(c)



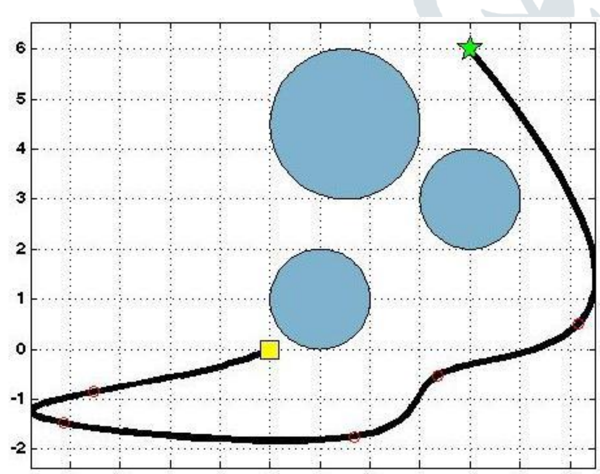
(d)



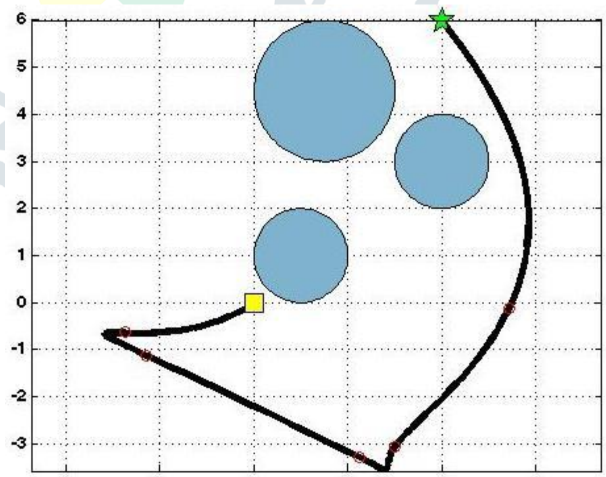
(e)



(f)



(g)



(h)

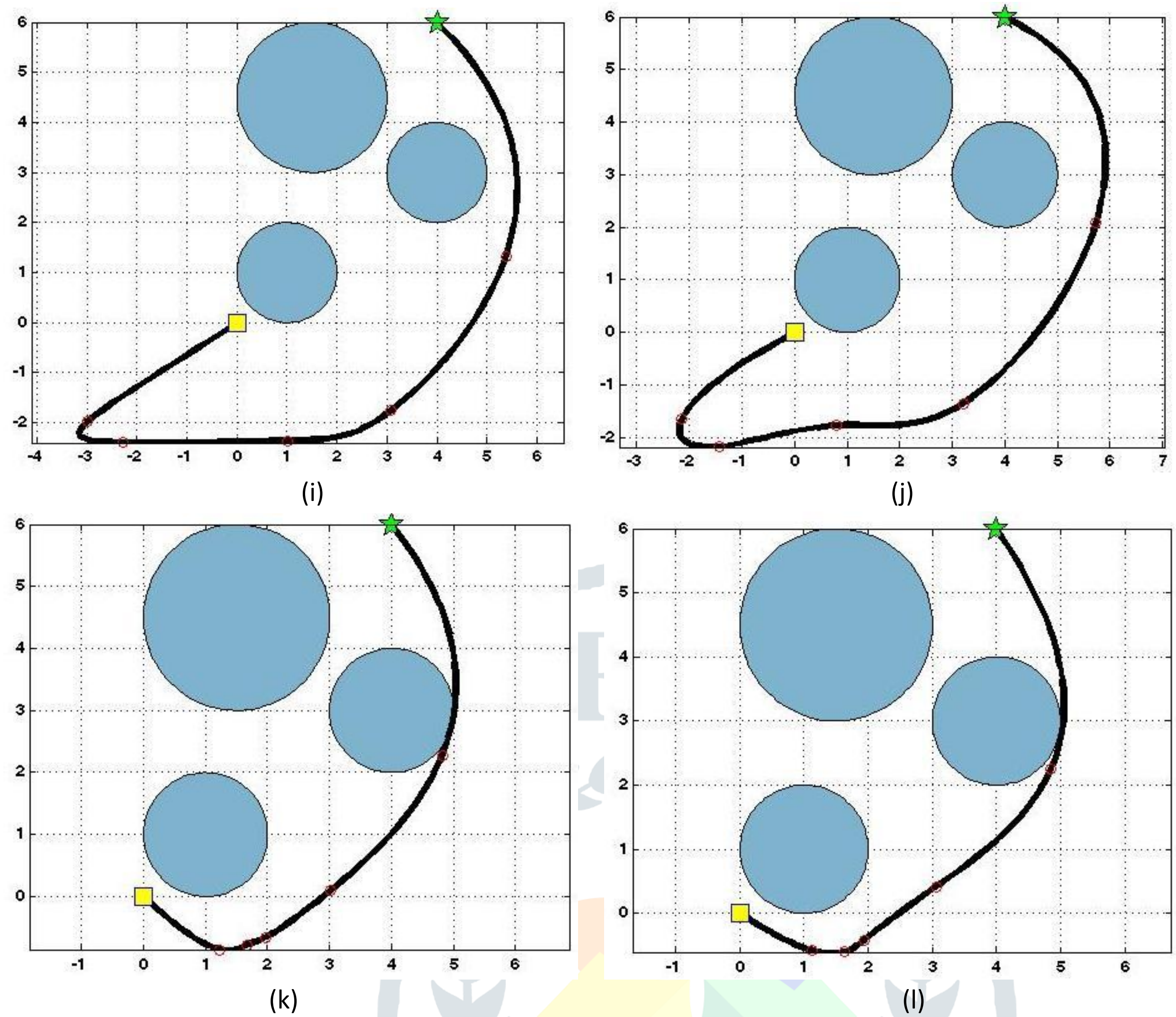


Fig. 4. The figure shows different stages of ACO for calculating Best Cost involved in finding Best Path from source (yellow box) to the destination (green star) with three obstacles (blue circles) on the way

Fig. 5 depicts the readings of Best Cost of the first twenty iterations involved in reaching from source to destination.

```
Iteration 1: Best Cost = 60.5696, 0.0017157
Iteration 2: Best Cost = 54.6147 @
Iteration 3: Best Cost = 48.057 @
Iteration 4: Best Cost = 28.3514 @
Iteration 5: Best Cost = 28.3514 @
Iteration 6: Best Cost = 28.3514 @
Iteration 7: Best Cost = 24.6034 @
Iteration 8: Best Cost = 24.6034 @
Iteration 9: Best Cost = 24.0617 @
Iteration 10: Best Cost = 24.0617 @
Iteration 11: Best Cost = 23.674 @
Iteration 12: Best Cost = 23.0388 @
Iteration 13: Best Cost = 21.0608 @
Iteration 14: Best Cost = 19.4088 @
Iteration 15: Best Cost = 19.4088 @
Iteration 16: Best Cost = 19.4088 @
Iteration 17: Best Cost = 17.7442 @
Iteration 18: Best Cost = 14.5125 @
Iteration 19: Best Cost = 14.5125 @
Iteration 20: Best Cost = 13.9624 @
```

Fig. 5. The figure depicts the readings of Best Cost of the first twenty iterations involved in reaching from source to destination.

Fig. 6 depicts the readings of Best Cost of the last twenty iterations involved in reaching from source to destination.

```
Iteration 20: Best Cost = 13.9624 @
Iteration 21: Best Cost = 13.9624 @
Iteration 22: Best Cost = 13.9624 @
Iteration 23: Best Cost = 12.8983 @
Iteration 24: Best Cost = 12.7167 @
Iteration 25: Best Cost = 12.7167 @
Iteration 26: Best Cost = 12.7167 @
Iteration 27: Best Cost = 12.5455 @
Iteration 28: Best Cost = 11.7216 @
Iteration 29: Best Cost = 10.9613 @
Iteration 30: Best Cost = 10.7649 @
Iteration 31: Best Cost = 10.6815, 6.5187e-05
Iteration 32: Best Cost = 10.6815, 6.5187e-05
Iteration 33: Best Cost = 10.583 @
Iteration 34: Best Cost = 10.4944 @
Iteration 35: Best Cost = 10.4944 @
Iteration 36: Best Cost = 10.4944 @
Iteration 37: Best Cost = 10.4779 @
Iteration 38: Best Cost = 10.4612 @
Iteration 39: Best Cost = 10.3809 @
Iteration 40: Best Cost = 10.1664 @
```

Fig. 6. The figure depicts the readings of Best Cost of first twenty iterations involved in reaching from source to destination

Fig. 7 shows the plotted graph based on the readings of forty iterations as per Case 2. The X-axis represents the number of iterations and Y-axis represents the obtained Best Cost.

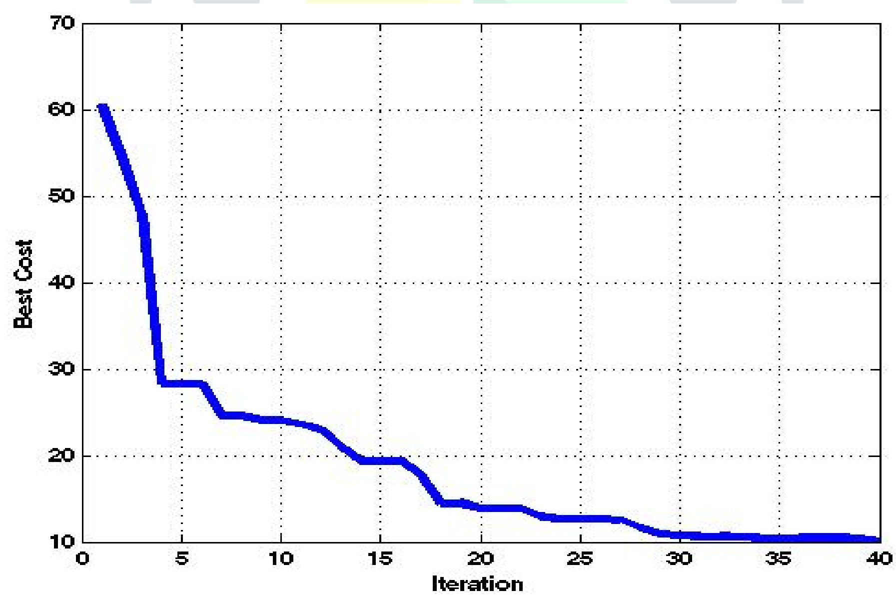


Fig. 7. The figure shows the plotted graph based on the readings of forty iterations as per Case 2

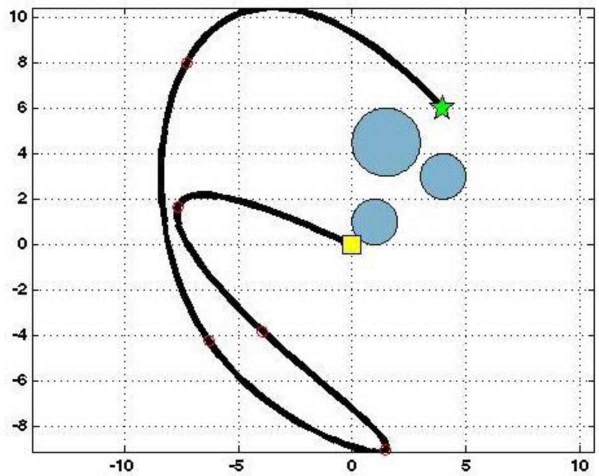
Case 3:

The readings of different parameters in case 1 are mentioned as under.

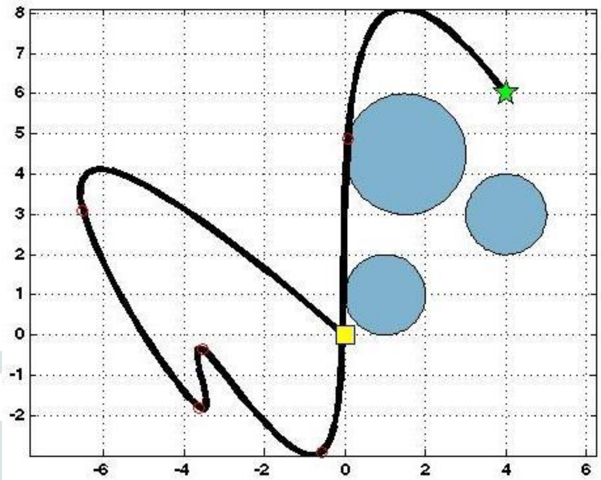
- Maximum Number of Iterations - 60
- Population Size (Swarm Size) - 2
- Inertia Weight - 1

- Inertia Weight Damping Ratio - .98
- Personal Learning Coefficient - 1.5
- Global Learning Coefficient - 1.5
- Number of handle points - 5

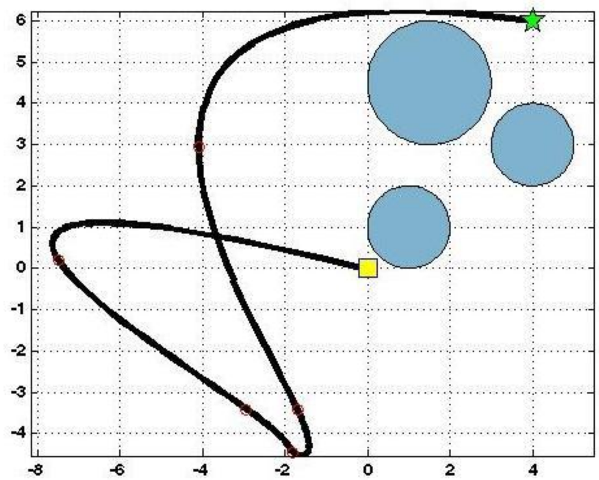
The implementation of ACO as per parameter values mentioned in Case 3 is shown in Fig. 8.



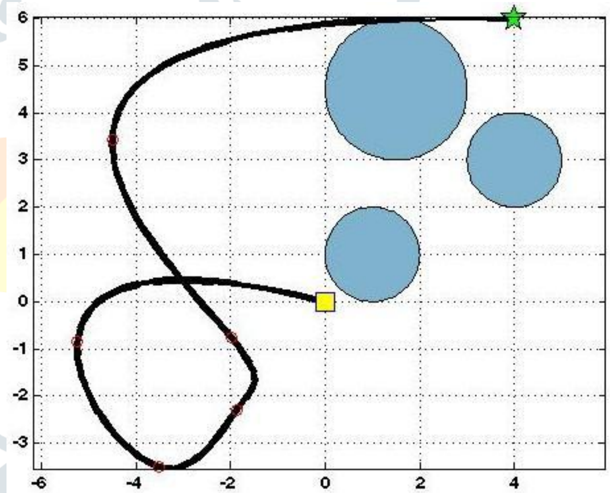
(a)



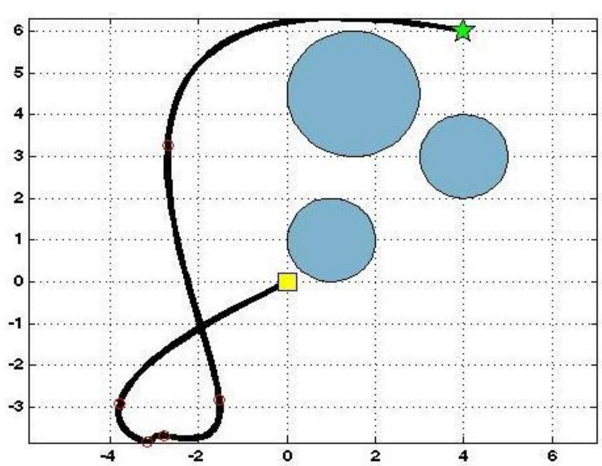
(b)



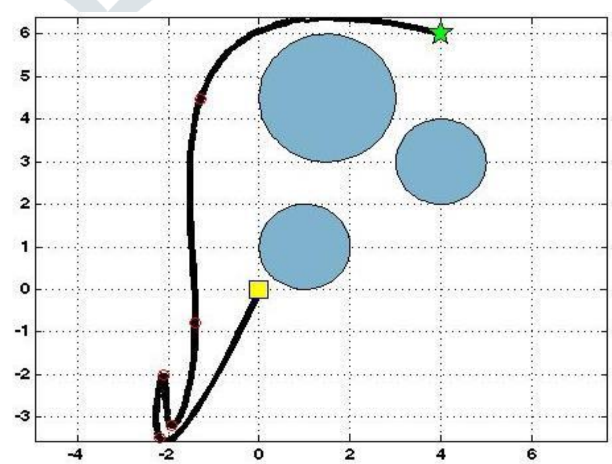
(c)



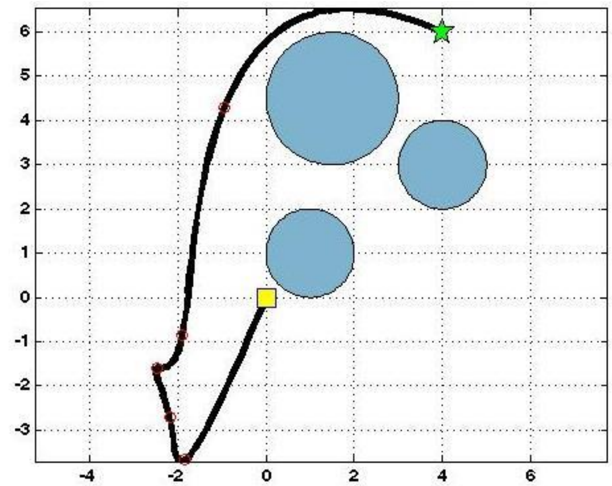
(d)



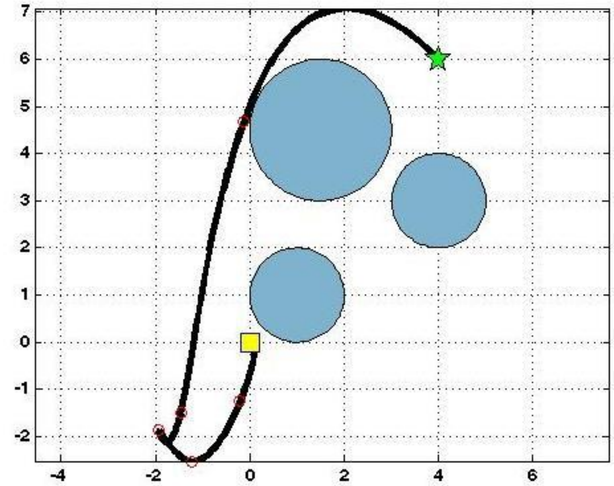
(e)



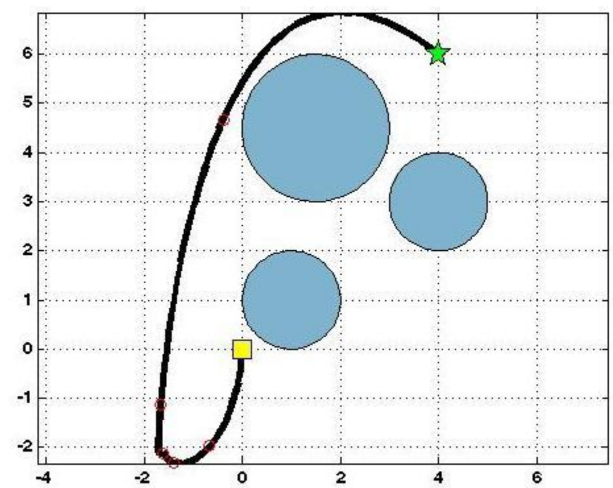
(f)


6 5 4 3 2 1 0 -1 -2 -3
-4 -2 0 2 4 6

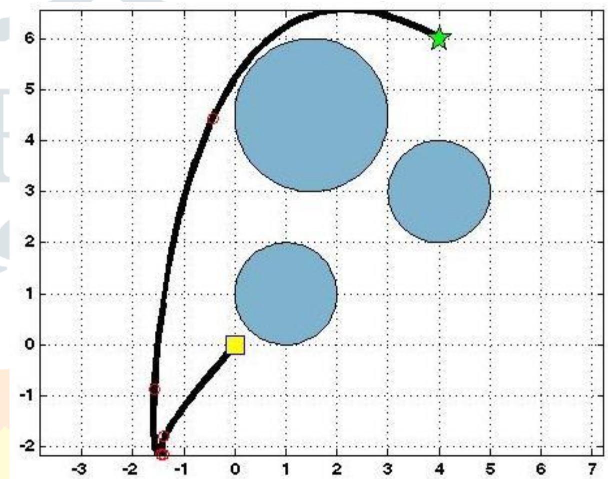
(g)


7 6 5 4 3 2 1 0 -1 -2
-4 -2 0 2 4 6

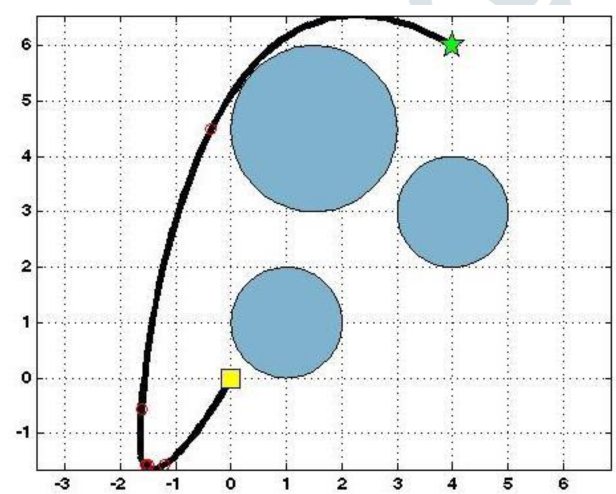
(h)


6 5 4 3 2 1 0 -1 -2
-4 -2 0 2 4 6

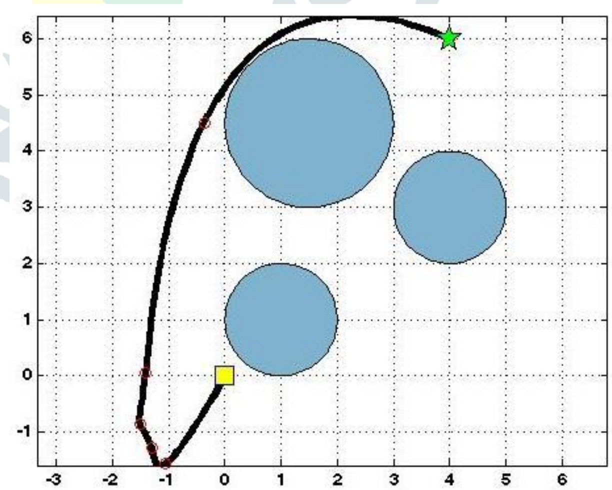
(i)


6 5 4 3 2 1 0 -1 -2
-3 -2 -1 0 1 2 3 4 5 6 7

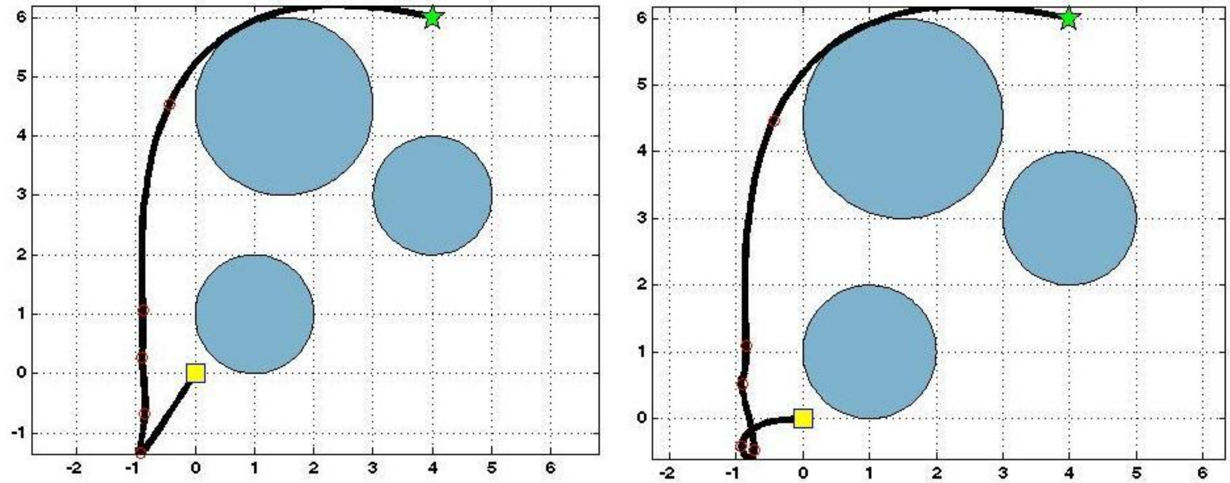
(j)


6 5 4 3 2 1 0 -1
-3 -2 -1 0 1 2 3 4 5 6

(k)


6 5 4 3 2 1 0 -1
-3 -2 -1 0 1 2 3 4 5 6

(l)



(m) (n)

Fig. 8. The figure shows different stages of ACO for calculating Best Cost involved in finding Best Path from source (yellow box) to the destination (green star) with three obstacles (blue circles) on the way

Fig. 9 depicts the readings of Best Cost of the first twenty iterations involved in reaching from source to destination.

```
Iteration 1: Best Cost = 58.616 @
Iteration 2: Best Cost = 58.616 @
Iteration 3: Best Cost = 36.7709, 0.00036529
Iteration 4: Best Cost = 34.0738 @
Iteration 5: Best Cost = 34.0738 @
Iteration 6: Best Cost = 28.1112 @
Iteration 7: Best Cost = 27.7821 @
Iteration 8: Best Cost = 27.7821 @
Iteration 9: Best Cost = 23.1859 @
Iteration 10: Best Cost = 21.7199 @
Iteration 11: Best Cost = 21.7199 @
Iteration 12: Best Cost = 21.7199 @
Iteration 13: Best Cost = 21.1309 @
Iteration 14: Best Cost = 21.1309 @
Iteration 15: Best Cost = 21.0881 @
Iteration 16: Best Cost = 18.7785 @
Iteration 17: Best Cost = 18.7785 @
Iteration 18: Best Cost = 18.7785 @
Iteration 19: Best Cost = 17.9506 @
Iteration 20: Best Cost = 17.9506 @
```

Fig. 9. The figure depicts the readings of Best Cost of first twenty iterations involved in reaching from source to destination

Fig. 10 depicts the readings of Best Cost of the last twenty iterations involved in reaching from source to destination.

```
Iteration 40: Best Cost = 13.3442 @
Iteration 41: Best Cost = 13.2892 @
Iteration 42: Best Cost = 13.2156 @
Iteration 43: Best Cost = 13.1735 @
Iteration 44: Best Cost = 13.1735 @
Iteration 45: Best Cost = 13.1217 @
Iteration 46: Best Cost = 13.1131, 2.5433e-05
Iteration 47: Best Cost = 13.1034 @
Iteration 48: Best Cost = 13.0892 @
Iteration 49: Best Cost = 13.0845 @
Iteration 50: Best Cost = 13.0673 @
Iteration 51: Best Cost = 13.0594 @
Iteration 52: Best Cost = 13.0041 @
Iteration 53: Best Cost = 12.9797 @
Iteration 54: Best Cost = 12.9559 @
Iteration 55: Best Cost = 12.8861 @
Iteration 56: Best Cost = 12.709 @
Iteration 57: Best Cost = 12.0217 @
Iteration 58: Best Cost = 11.8333, 5.4298e-06
Iteration 59: Best Cost = 11.7847 @
Iteration 60: Best Cost = 11.7143, 3.4326e-05
```

Fig. 10. The figure depicts the readings of Best Cost of last twenty iterations involved in reaching from source to destination

Fig. 11 shows the plotted graph based on the readings of sixty iterations as per Case 3. The X-axis represents the number of iterations and Y-axis represents the obtained Best Cost.

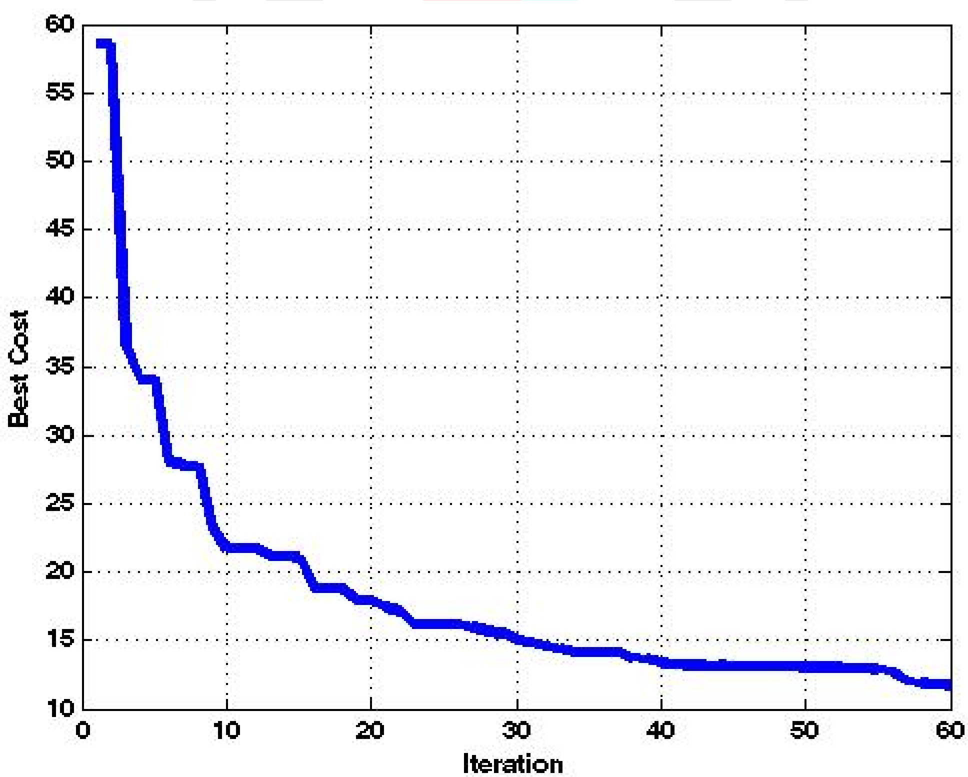


Fig. 11. The figure shows the plotted graph based on the readings of sixty iterations as per Case 3

III. CONCLUSION

The research paper discussed the working of the ACO algorithm with obstacles. The paper implemented three cases with a different number of iterations. The different stages adopted to reach from source to destination via Best Path at Best Cost have been pictorially shown in each of the three cases. The Best Cost readings of each intermediary step have been recorded to monitor the decrease in the value of Best Cost. In each case, the last reading of the value of Best Cost is lower as compared to the initial reading. This proves the worth of research conducted. In future work, the implementation can be made more dynamic by changing other parameters like the number of obstacles, local learning and global learning parameters.

REFERENCES

[1]. L.dos Santos Coelho and D. L. de Andrade Bernert, ''A modified ant colony optimization algorithm based on differential evolution for chaotic synchronization,'' Expert Syst. Appl., vol. 37, no. 6, pp. 4198–4203, 2010.

[2]. B. K. Nanda, G. Das, "Ant colony optimization: a computational intelligence technique", *Int. J. Comput. Commmun. Technol*, vol. 2, no. 6, pp. 105-110, 2011.

[3]. Deng, R. Chen, B. He, Y. Q. Liu, L. F. Yin, and J. H. Guo, ''A novel two-stage hybrid swarm intelligence optimization algorithm and application,'' Soft Comput., vol. 16, no. 10, pp. 1707–1722, 2012.

[4]. W. Deng, H. M. Zhao, L. Zou, G. Y. Li, X. H. Yang, and D. Q. Wu, ''A novel collaborative optimization algorithm in solving complex optimization problems,'' Soft Comput., vol. 21, no. 15, pp. 4387–4398, 2017.

[5]. Rada-Vilela, M. Chica, Ó. Cordón, and S. Damas, ''A comparative study of multi-objective ant colony optimization algorithms for the time and space assembly line balancing problem,'' Appl. Soft Comput., vol. 13, no. 11, pp. 4370–4382, 2013.

[6]. D. I. D. Ariyasingha and T. G. I. Fernando, ''Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem,'' Swarm Evol. Comput., vol. 23, pp. 11–26, Aug. 2015.

[7]. Krynicki, M. E. Houle, and J. Jaen, ''An efficient ant colony optimization strategy for the resolution of multi-class queries,'' Knowl.-Based Syst., vol. 105, pp. 96–106, Aug. 2016.

[8]. R.-H. Huang and T.-H. Yu, ''An effective ant colony optimization algorithm for multi-objective job-shop scheduling with equal-size lot-splitting,'' Appl. Soft Comput., vol. 57, pp. 642–656, Aug. 2017.

[9]. Z. Zhouet al., ''A multi-objective multi-population ant colony optimization for economic emission dispatch considering power system security,'' Appl. Math. Model., vol. 45, pp. 684–704, May 2017.

[10]. P. K. Tiwari and D. P. Vidyarthi, ''Improved auto control ant colony optimization using lazy ant approach for grid scheduling problem,'' Future Gener. Comput. Syst., vol. 60, pp. 78–89, Jul. 2016.

[11]. Akka and F. Khaber, ''Mobile robot path planning using an improved ant colony optimization,'' Int. J. Adv. Robot. Syst., vol. 15, no. 3, pp. 1–7,2018.

[12]. W. Deng, H. M. Zhao, X. H. Yang, J. Xiong, M. Sun, and B. Li, ''Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment,'' Appl. Soft Comput., vol. 59, pp. 288–302, Oct. 2017.