Bee-Inspired Algorithms and its implementation in reducing the transportation cost: Optimizing the Travelling Salesman Problem

IE 562 PROJECT REPORT

PALLAVI SHARMA

I. INTRODUCTION:

Travelling Salesman Problem (TSP) also called the Travelling Salesperson Problem is defined as the problem of finding the shortest possible route for a given set of cities, that visits each city and returns to the origin city. The Travelling Salesman Problem has found its implications in the field of reducing transportation costs in the supply chain. Being a NP-hard problem, using exact methods to acquire optimal solutions becomes difficult, thus metaheuristics have been used so far to find optimal solutions. The meta heuristic of Swarm Intelligence, inspired by the collective behavior of natural insect colonies and animal societies, uses the approximate and non-deterministic strategies to give near-optimal solutions for TSP. Artificial Bee Colony inspired by the swarm intelligence is one such algorithm that has been found to give optimal solutions for TSP. This project thus uses Artificial Bee Colony (ABC) algorithm to solve the Travelling Salesman Problem (TSP). It then progresses into modifying the Artificial Bee Colony (ABC) algorithm by combining it with dynamic programming to get a better optimal solution with **25% improved cost** and an improved computational efficiency of the algorithm.

II. ARTIFICIAL BEE COLONY (ABC) ALGORITHM

This algorithm was first coined by the Karaboga [1] in 2005 for optimizing the numerical problems, further modified by Banharnsakun et al [2] to solve the travelling salesman problem. This algorithm being inspired by the foraging behavior of the honeybee colonies consists of three types of bees: the employed bees, the onlooker bees and the scout bees. The employed bees go out to exploit the found food sources and look out for better food sources nearby. These employed bees after returning to the hive perform waggle dance to communicate the richness of the food source, information about the direction and distance to patches of flowers yielding nectar and pollen, to water sources, or to near-site locations. The onlooker bees seeing the waggle dance choose an employed bee's solution to further exploit it. If a bee is not able to find an optimal solution after repetitive iterations, it becomes a scout bee and thus abandons the path discovered by it and starts finding a new food source [3]. In the proposed Artificial Bee Colony (ABC) algorithm in this project, roulette wheel has been used to select the employed bee for the onlooker bee to follow, wherein the quality of the food available is used as the fitness function for the process as mentioned in [4].

II.A. APPROACH FOLLOWED

For the purpose of solving the Travelling Salesman Problem (TSP), the employed bees search for the best optimal route to cover all the cities in the least distance possible, these bees after returning to the hive communicate

their knowledge with the onlooker bees through their waggle dance. The roulette wheel selection method has been used to select the employed bee for the onlooker bee to follow, depending on the fitness of the bee, which is calculated as a function of the Euclidean distance calculated between the subsequent cities traversed in the route. Furthermore, the mutation operation of Genetic Algorithm (GA) was used to mutate the employed bees in search of optimizing the paths followed. The scout bees were abandoned if the number of trials for the bees exceeded the maximum number of trials defined for the bees. The Cost function being used for Travelling Salesman Problem can be defined as:
For a set of N cities defined as a bee, the Euclidean distance will be calculated as:

$$= \sum_{i=0}^{N-2} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_i + 1)^2}$$
$$+ \sqrt{(x_{N-1} - x_0)^2 + (y_{N-1} - y_0)^2}$$

(x,y) are the coordinates of the 50 cities for which the distance will be calculated. First sum of all the distances from the initial city in the route to the last city is calculated and then distance from the last city in the route to the initial city is calculated, to find the total distance traversed in the Travelling Salesman Problem (TSP).
The Brute-Force approach was applied with the different number of bees and iterations to find the best optimal solution. The figure 1 shows the dips and ups in the graph plotted for the 10 runs of a combination of 100 bees and 600 iterations, with the cost varying from **1145.66** to **1276.20**.

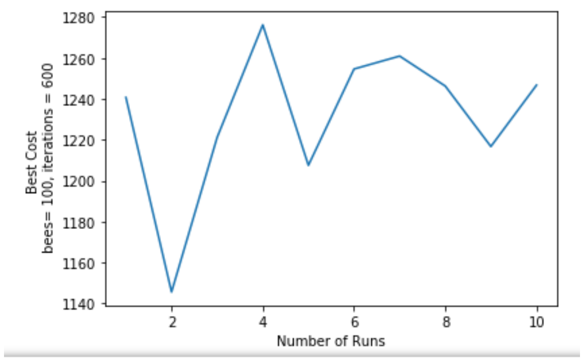


Fig. 1 Bees = 100, Iterations = 600

II.B <u>DESIGN OF EXPERIMENTS</u>

The different factors that can be varied in the solution are the number of bees employed to find the best solution and the number of iterations for each bee, wherein in each iteration the bees were mutated, cost functions were calculated to find the best solution in each iteration. Thus, the costs and times were calculated by:

1. Varying the number of iterations and keeping the number of bees fixed
2. Varying the number of bees and keeping the number of iterations fixed.

<u>RESULTS</u>:

A. <u>Fixed Number of Iterations and Increasing the number of bees</u>

The number of bees were increased from 50 to 600 for each iteration value ranging from 50 to 600. Also, 10 runs were made for each combination to find the best optimal cost value of the Travelling Salesman Problem. The following figures show the best cost resulted for the different number of bees when the iterations were kept constant. Thus, as seen in figure 2, with 50 iterations the best cost varied from 1340.39 for 100 bees to 1245.67 for 600 bees.

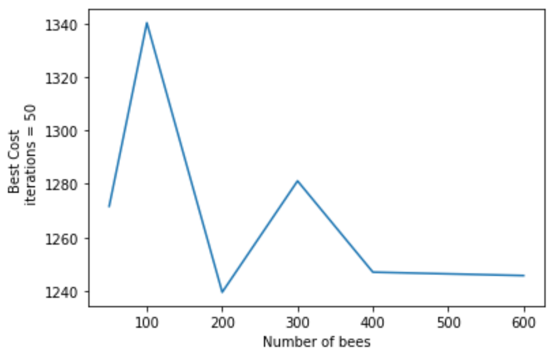


Fig.2 50 Iterations, variable number of bees

Furthermore, when seen for 600 iterations and different number of bees in the figure 3, the cost varied from 1225 for 50 bees to 1138.2139 for 600 bees.

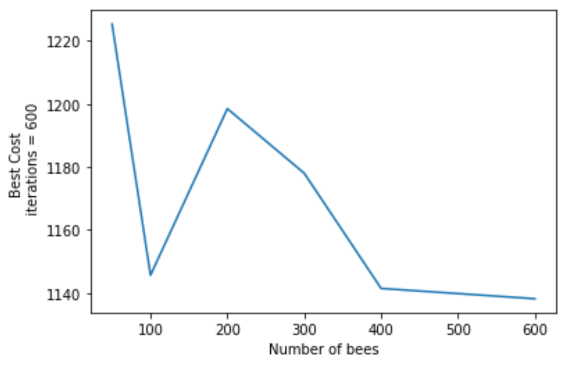


Fig.3 600 iterations and variable number of bees

B. <u>Fixing number of bees and varying iterations</u>

By fixing the number of bees and increasing the number of iterations, a general decreasing trend was seen in the cost of the problem as seen in figure 4 for 100 bees and in figure 5 for 400 bees. It can be inferred from the graphs that the Artificial Bee Colony (ABC) Algorithm is based on randomization to explore the solution space. This is because in the Artificial Bee Colony (ABC) Algorithm, the bees are generated randomly and the mutation taking place to mutate the employed bees also generates random results.

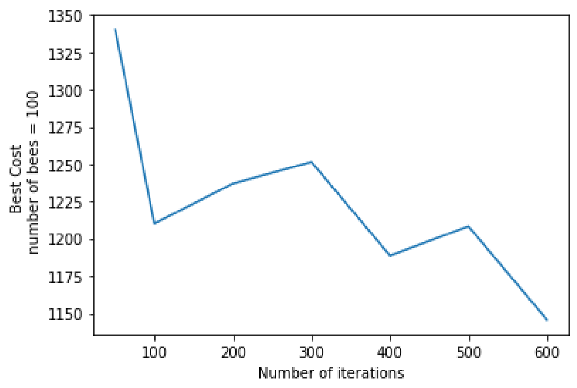


Fig.4 Number of bees = 100; different iterations

There is no guarantee of the cost converging to a global optimum, as can be seen through the variable dips and ups in the graphs. Thus, Artificial Bee Colony Algorithm can still be improvised to give better optimal results, which will be seen in the following section.

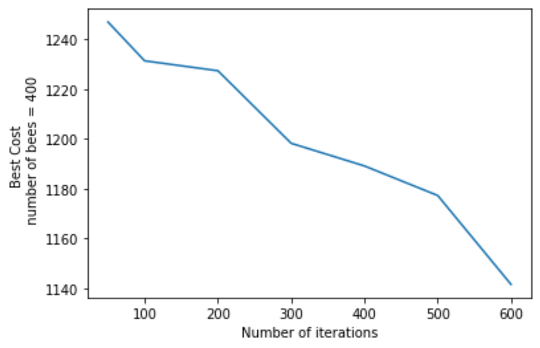


Fig.5 Number of bees = 400; different iterations

II.C COMPUTATIONAL EFFICIENCY

For the Travelling Salesman Problem (TSP), we have taken 51 cities to be traversed. For N number of cities, the different number of combinations of routes that can be explored are N!. Thus, in our case it will be 51!. With the 600 bees and 200 iterations, the total number of possible solutions that are explored are 1,20,000 which is 7.7363e^{-60} % which is not even close to 1% of the total space to be explored. For a reduced number of 10 cities, the total possible solutions are 10! If we take 600 bees and 200 iterations to find optimal solution for this set, only 3.30% of the total solution space will be explored.

These numbers indicate that brute force approach of Artificial Bee Colony (ABC) Algorithm covers only a small fraction of total solution space. An increased number of bees or iterations has a greater chance of increased computation. Thus, to find the global optimal solution the number of computations required will be of the order **O(n!)** which in the world of optimization is very slow and unsustainable. With an interest in improving the computational efficiency of the algorithm along with obtaining an even optimal solution, a combination of dynamic programming and Artificial Bee Colony Algorithm has been explored as a part of this project.

III MODIFIED ARTIFICIAL BEE COLONY ALGORITHM

Inspired by the approach of Dynamic Programming, methods of optimizing the artificial bee colony algorithm has been explored. While creating an initial bee population, the bees are allocated random routes. These bees are further mutated to result into solutions which may or may not be better than the original solution. The intuition behind the approach is to make each bee follow the same sub-path as the best bee in each iteration, followed by mutation to find the next vertex moving the current solution towards optimum. This can be better explained as: At the start of iteration i, for each bee the sub-path containing (i-1) vertices will be the same, and mutation is then performed on each bee for the remaining positions, from ith vertex to the last vertex. At the end of this iteration i, best bee is found, and we fix the ith index for each bee as the vertex at the ith index of the best bee in the current iteration. This process is repeated n times, for the length of the path.

Similar to the original Artificial Bee Colony Algorithm, roulette wheel selection method is used to choose the employed bee for the onlooker bees to follow, based on the quality or fitness of the employed bees which is calculated in terms of the Euclidean distance traversed from the initial city to the ending city and back to initial city.

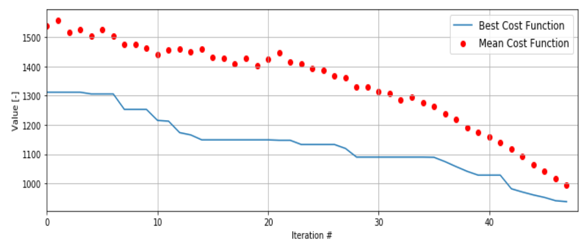
III.A. COMPUTATIONAL EFFICIENCY

Since, in each iteration an index in each bee kept on getting fixed, the total number of steps to reach a local optimum solution is a linear function of the total number of cities to be explored. Thus, the computational efficiency of the algorithm decreased to **O(n),** which is a significant improvement in the total computational efficiency of the algorithm from factorial computation to a linear computation. Also, since the indices of the bees were fixed in each iteration, the total number of iterations for any number of bees reduced to the number of cities present in a bee object. This further improved the time efficiency of the algorithm since as opposed to original algorithm wherein with the increase in the number of iterations, the results improved, in this case with the fixed number of iterations, a better result was seen.

Although this solution may not converge to a global optimum, however, it is seen to perform better than the brute force approach in linear O(n) time as compared to O(n!) for Artificial Bee Colony algorithm.

III.B. RESULTS

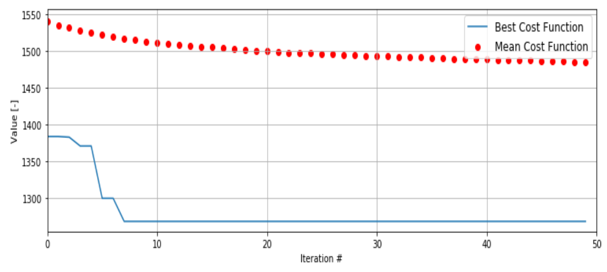
The Modified Artificial Bee Colony Algorithm showed improved results for different number of bees. With 600 bees, the best solution obtained is 937.29 (figure 6) in a time span of 3212 seconds, opposed to the cost of 1245.6797 (figure 7) in a time span of 2708.92 seconds. An improvement of **24.75%** can be seen in the cost of the Travelling Salesman Problem (TSP).



```
Fitness Value ABC: 937.2920974548877
Bee Hive done! Time elapsed: 3212.8491730690002 seconds
```

Fig.6 600 bees (Modified Artificial Bee Colony Algorithm)

Similarly, for 300 bees, the cost is 1018.855 as opposed to 1270.0132 for original Artificial Bee Colony (ABC) Algorithm, showing an improvement of **19.77%** in the cost.



```
Fitness Value ABC: 1268.0530636417263
Bee Hive done! Time elapsed: 3551.1436936855316 seconds
```

Fig.7 600 bees (Original Artificial Bee Colony Algorithm)

The following table 1 shows the comparison of the different cost values for Artificial Bee Colony (ABC) algorithm and for modified Artificial Bee Colony (ABC) Algorithm. Thus, the modified Artificial Bee Colony (ABC) Algorithm proved to give approximately **25%** better optimal solution than the original Artificial Bee Colony Algorithm, along with reducing the computational efficiency of the algorithm from O(n!) to O(n).

In the figure 8, the best cost can be seen varying from 1262.0167 for 50 bees to 937.29 for 600 bees, showing an improvement of 25% cost with an increase in the number of bees.

BEES	OPTIMISED COST	OPTIMISED TIME	ABC COST	ABC TIME
50	1262.0167	39.428501	1271.7	37.258
100	1078.1787	310.99505	1340.4	82.605
200	1035.2221	321.82977	1239.4	182.47
300	1018.8551	941.23711	1270	1483.6
400	1012.3489	1632.3698	1247	1164.207
500	938.34857	2764.7882	1263.5	1461.3
600	937.2921	3212.8492	1245.7	2708.9

Table 1 Modified ABC vs original ABC

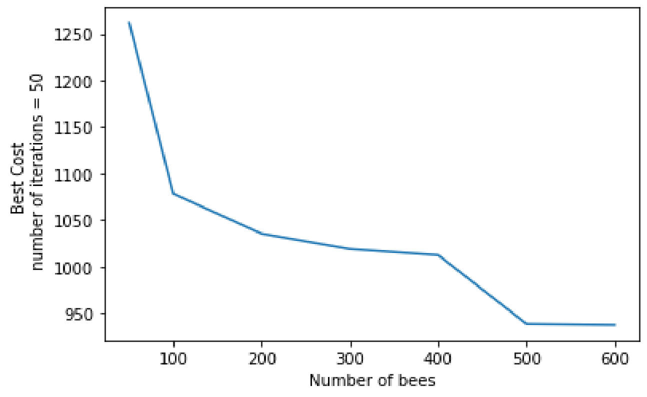


Fig.8 Modified Bee Colony Algorithm for different number of bees

IV. DIFFICULTIES FACED

One of the main difficulties faced is the amount of time required for completion of the algorithm on a single machine as the combination of number of bees and iterations goes higher in search of an optimal solution. Even parallelizing the algorithm and running it on multiple machines will only give a very small improvement considering the large size of the solution space. Another difficulty faced is not having a control over the randomization function to generate unique permutations in each mutation.

V. FUTURE SCOPE

The optimized Artificial Bee Colony (ABC) algorithm can be improved by having a large number of bees exploring a large number of permutations for each iteration. This large swarm of bees can leverage modern High-Power Computing (HPCs) facilities or can be distributed across several machines in order to decrease the running time and increase the total percent of solution space covered. Another improvement that can be made is in the initialization step, by fixing the origin city for each bee with different cities, so as to have a different starting point for each bee. In the current initialization phase, the bees are randomly generated wherein there is a possibility for most of the bees to have the same starting point and that the most optimal starting point is missing from the complete bee population. This approach will more closely resemble the Dynamic Programming solution, thus bringing it near to an even optimal solution.

VI. REFERENCES

1. http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm

2. A. Banharnsakun, T. Achalakul, and B. Sirinaovakul, "ABC-GSX: A hybrid method for solving the traveling salesman problem," in *Proceedings of the 2010 2nd World Congress on Nature and Biologically Inspired Computing, NaBIC 2010*, pp. 7–12, jpn, December 2010
3. Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem- Shuzhu Zhang, C.K.M.Lee, K.L.Choy, WilliamHo, W.H.Ip

4. D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, pp. 21–57, 2014
5. https://en.wikipedia.org/wiki/Travelling_salesman_problem

VI. APPENDIX

The Python codes for the Artificial Bee Colony (ABC) Algorithm and Modified Artificial Bee Colony (ABC) Algorithm have been attached as Jupyter notebooks with the report.