# Travelling Salesman Problem using Bee Colony with SPV

**Nishant Pathak, Sudhanshu Prakash Tiwari**

***Abstract:*** ***Challenge of finding the shortest route visiting each member of a collection of locations and returning to starting point is an NP-hard problem. It is also known as Traveling salesman problem, TSP is specific problem of combinatorial optimization studied in computer science and mathematical applications. In our work we present a solution for TSP problem using ABC with SPV rule. In this method we extend Artificial Bee Colony algorithm using SPV rule. Artificial bee colony algorithm solves real coded optimization problems but travelling salesman problem is a discrete optimization problem for converting the ABC algorithm to solve TSP problem SPV rule is used. Artificial Bee Colony (ABC) is an optimization algorithm based on the intelligent behavior of honey bee swarm. In the proposed method we extend ABC with SPV rule for local search strategy. The experimental results show that our proposed ABC with SPV performs better than GA (Genetic Algorithm), our ABC with SPV model can reach broader domains in the search space and show improvements in both precision and computational time.***

***Index Terms*****:** ***ABC, Artificial Bee Colony, GA, Genetic Algorithm, TSP, SPV.***

## I. INTRODUCTION

The selection of a best element from some set of available alternatives referred as optimization problem. In many such problems, exhaustive search is not feasible. It has important applications in several fields, including artificial intelligence, machine learning, mathematics, and software engineering. Most of such problems are considered NP-hard i:e they cannot be solved to optimality in polynomial computational time. Some common problems involving optimization are the traveling salesman problem, the minimum spanning tree problem etc.

The travelling salesman problem (TSP) is an NP-hard problem in combinatorial optimization studied in operations research and theoretical computer science.TSP has several applications, such as planning, logistics, network communication, transportation, and the manufacture of microchips.

TSP states that for one salesman who wants to visit cities and given distances between them. Travelling salesman has to visit all of them, but he does not to travel very much. Task is to find a sequence of cities to minimize travelled distance [1][16].

There are many literature available pertaining to this problem. Homer's Ulysses in his work [3] attempts to visit the cities described in the Odyssey exactly once, there are 653,837,184,000 distinct routes. To calculate an optimal route requires 92 hours on a powerful 28 MIPS workstation. An instance with 85,900 points was solved using Concorde TSP Solver, taking over 136 CPU-years [1].

There are various metaheuristics and approximation algorithms, which quickly yield good solutions, have been devised [4][5][6]. Modern methods can find solutions for extremely large problems (millions of cities) within a reasonable time. Several algorithms employing metaheuristic approaches such as Genetic Algorithm (GA) [4], Ant Colony Optimization (ACO) [9], Particle Swarm Optimization (PSO) [6] [17] or Bee Colony Optimization (BCO) [5][7] were applied to solve TSP [2] [14] [15].There are few hybridized method also present which solves the TSP. Sarayut and Siripom [8] proposed ACO-GA and ACO-PSO which are the hybrid methods. These methods perform based on the adjustment of a parameter Qo. If Qo is high, the local search would be geared towards exploitation, and when Qo is low, the search is geared towards exploration. Wong, et al. [7] applied BCO to solve TSP. These algorithms yield solutions with lower precisions on average. To improve upon these results the ABC algorithm [10],an effective algorithm already being applied to several optimization applications [11], is focused.

In this paper we solve the travelling salesman problem using the real version of ABC algorithm using SPV rule. SPV rule is used to generate the routing sequence using the real value generated by the artificial bee colony algorithm. To check the efficiency of our proposed solution we compared our solution with the solution of genetic algorithm [12] [13] [18].

The organization of the paper is as follows section II gives brief introduction on Artificial Bee Colony. Section III gives the introduction of TSP. Section IV describes the Proposed Methodology. Section V outlines the Experimental setup and result, Section VI gives Conclusion.

## II. ARTIFICIAL BEE COLONY

The Artificial Bee Colonies (ABC) is another novel optimization algorithm that comes under Swarm Intelligence. ABC algorithm is inspired by social behavior of natural bees. The Artificial Bee Colony, introduced by Dervis Karaboga in 2005 (Karaboga, 2005; Karaboga & Basturk, 2007) [10] [11].

**Manuscript received on July, 2012**.

**Nishant Pathak,** Lovely Professional University Phagwara, Punjab
**Sudhanshu Prakash Tiwari** Lovely Professional University Phagwara, Punjab**.**

The natural bees are very good in searching for some food source. Whenever any bee finds the food, it signals the other bees by its dance. This signals the other bees regarding the quantity and the location of the food source. This helps in directing the other bees towards good sources of food in their search for food. These bees are able to attract a large number of bees and carry forward to search the food area. In ABC algorithm the bees are divided into three groups: employed bees, onlooker bees and scout bees. Employed bees visited the food source and gather information about food source location and the quality. Employed bees have memory, so they know the places they have visited before and the quality of food there. Employed bees performs the local search and try to exploit the neighboring locations of the food source and search the best places of foods in the surrounding areas of the present value. Onlooker bees are bees that are waiting on the dance area to decide which food source is better. This decision is made on the basis of information provided by employed bees. Onlooker bees perform the global search for discovering the global optimum. Scout bees do a random search for the food. Scout bees discovers the new area which are uncovered by the employed bees, these bees are completely random in nature and their operation of search. Scout bees avoid the search process to get trapped in local minima. In ABC algorithm, first half of the colony consists of employed bees and the second half constitutes by onlooker bees. The number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source is exhausted becomes the scout bees.

In ABC algorithm each food source position represents a candidate solution of optimization problem. In optimization problem each solution is associated with the fitness value on the basis of fitness value it is decided that which solution is better. So the nectar amount of a food source corresponds to the fitness value of the associated solution in ABC algorithm. The number of employed bees or the onlooker bees is equal to the number of solutions in the population. The ABC algorithm generates a random solution or initial population of size NF, where NF denotes the size of population or total number of food source. Each solution is represents the position of food source and denoted as $x_{ij}$, where i represents a particular solution (i=1,2,…..,NF) and each solution is a D-dimensional vector so j represents a particular dimension of a particular solution (j=1,2,….,D). After initialization of random solution employed bees start their searching. Employed bees search the food source near the previous food source, if the generated new solution is better than the previous solution than new solution replaces the old one. The comparison of food sources or solutions is done on the basis of fitness value or nectar amount of food source.

After all employed bees complete the search process; they share the nectar information of food sources (solutions) and their position information with onlooker bees. Now onlooker bee chooses a food source depending on the probability value $P_i$ associated with the food source. Probability value for each food source is calculated by following equation (1):

$$P_i = \frac{f_i}{\sum_{n=1}^{NF} f_n} (1)$$

Where $f_i$ is the fitness value of the solution *i* or the nectar amount of food source evaluated by employed bee and *NF* is the number of food source. So after the evaluation of the food source by the employed bees the probability value for each food source is determined which is used by onlooker bees.

To produce the candidate solution from the previous solution artificial bee uses the following equation (2):

$$v_{ij}= x_{ij}+\O_{ij}(x_{ij}-x_{kj}) \quad (2)$$

Where *j* is a index for dimension (*j=1,2,….,D*), *k* is a index which represents particular individual or solution from the population (*k=1,2,3…..,NF*), and *i* is also a index represents a particular solution (*i=1,2,…..,NF*). The difference between *i* and *k* is that *k* is determined randomly and value of *k* has to be different from *i*. $\O_{ij}$ is a random number between [-1,1]. It controls the production of the neighbor food positions around $x_{ij}$.. The difference between the parameters of the $x_{ij}$ and $x_{kj}$ decreases, the perturbation on the position $x_{ij}$ decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is reduced. After the production of candidate solution $v_{ij}$ , its fitness value is calculated and then it is compared with the fitness of $x_{ij}$. If the new candidate solution has equal or better nectar or fitness than the old, it is replaced with the old one in the memory. Otherwise, the old is retained. If a solution is not improved further through a predetermined number of cycles then that food source is assumed to be exhausted. Exhausted food source is replaced by new food source generated by scout bees.

## III. TRAVELING SALESMAN PROBLEM

Given a collection of cities and the cost of travel between each pair of them, the traveling salesman problem, or TSP for short, is to find the cheapest way of visiting all of the cities and returning to your starting point. In the standard version we study, the travel costs are symmetric in the sense that traveling from city X to city Y costs just as much as traveling from Y to X.

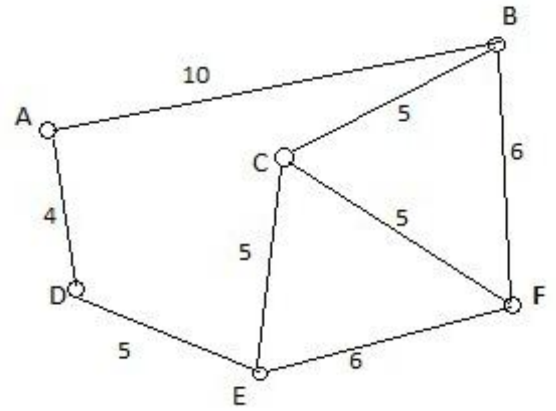

**Figure1: Graph Representation of TSP**

In figure 1 each node in graph represents a city and each weighted edges in graph represents path from one city to other city. Weight is the cost to travel to reach city. Fitness function of the TSP will be

$$T_c= \sum_{k=1}^{d} Cij \quad (3)$$

Where $C_{ij}$ is cost associated of path *i* to *j* and *k* is for the number of city or dimension.

## IV. PROPOSED METHODOLOGY

In this paper we have proposed a solution for Traveling Salesman problem using artificial bee colony algorithm with SPV rule. For solving any optimization problem we have to first formulate the problem according to optimization problem. In this case first we formulate the Traveling salesman problem according to our proposed ABC with SPV. Next subsection describes how we formulate the Traveling salesman problem.

### A. Representation

To solve the problem, representation of the individual and fitness value is required. Real coded artificial bee colony algorithm is based on population (candidate solution) and each individual have its own fitness value according to which it is compared from others, so we have to first represent the Traveling salesman problem in terms of ABC with SPV rule.

In TSP, we have a set of city, path from one city to other and cost associated to the path as input, which informs that which path is to be operated on which city and in which order as output.ABC with SPV is based on population concept and each individual in population represents a solution, in case of TSP, solution is a sequence of path traversing each city with minimum cost. So we have to first formulate each individual of ABC with SPV. Our proposed ABC algorithm generates the real value vector for an individual but the TSP problem is a discrete problem so we have to map the generated real value vector in to discrete values that used for a TSP problem. We have used a SPV rule (shortest position value) to map the generated real value vector to discrete value vector. For every generated individual $T_{id}$ of ABC we have generated another vector of discrete value $S_{ij}$ called sequence vector using SPV rule. The output for TSP problem is an optimal set of sequence of cities through which salesman has to travel. The sequence vector $S_{ij}$ generated for an individual used as sequence for salesman to travel the cities.We represent dimension of an individual as a number of city. The dimension value of each individual contains the real values from the search space. Each individual is represented by $T_{id} = \{t_{i1},t_{i2,}t_{i3},\ldots\ldots t_{id}\}$ andfor each individual there is a Sequence vector $S_{id}= \{Si_1,Si_2,Si_3,\ldots\ldots Si_d\}$ where *i* is the particular individual and *d* represents the dimension index, is calculated using ABC with SPV rule. Sequence vector of each individual makes transformation on the basis of individual real value vector $T_{id} = \{t_1,t_{2,}t_3,\ldots\ldots t_d\}$.The individual gets it value random initially and the associated sequence vector is transformed by using SPV rule on individual. The individual or real value vector has real value which is transformed to discrete value. Smallest position value i.e. SPV rule is used to find a permutation corresponding position $T_{id}$. The Individual $T_{id}$ has continuous values. By using the SPV rule this continuous position value can be converted to discrete value permutation $S_{id}= [s_{i1}, s_{i2},\ldots s_{id}]$. $S_{id}$ is the sequence of path of *I* individual in the processing order with respect to the *d* dimension. Randomly generated individuals are then being updated by employed bee and onlooker bees of Bee colony algorithm and again the transformation of updated particles to the sequence vector is done. After each modification of an individual the sequence vector associated with that individual is calculated.

For ex. if we have 6cities in Figure 1 then we have dimension value as 6. Based on SPV rules, the continuous position or individual convert to a permutation of sequences $S_{id}$, which is a sequence of city implied by the individual $T_{id}$. Individual $T_{id}$ is calculated using ABC= {4.83,-0.55, 1.90, 3.46, 1.05, and 2.87}

Then using SPV rule transformation of $T_{id}$ to $S_{id}$, we have $S_{id}$ value as {5, 0, 2, 4, 1, 3}. These values are generated using the values of the vector $T_{id}$ the shortest value is consider as the first value or first city i.e. city 0 and kept in a vector which will be used for accessing cost associated to path which is stored in 2 dimensional arrays in priory. All the values of sequence vector is calculated according the value generated by ABC.

### B. Fitness Function

After representation of each individual we have to calculate fitness value of each individual. On the basis of fitness value we determine the optimal solution. In case of TSP optimal solution is the minimization the value of equation (3). Equation (3) is the addition of the travelling cost from one city to another in a path to travel the whole city. For calculating the fitness value first we have to calculate the cost matrix. Cost matrix contains the cost of travel from one city to another.

Our main objective is to minimize the fitness value, an individual who have the minimum fitness value is considered as the optimal solution.

### C. Algorithm: Real Genetic Artificial Bee Colony

To solve the Traveling salesman problem we have used the Artificial Bee Colony Algorithm. We set an initial population by selecting random starting values from the search space than the sequence vector associated with each individual is calculated; sequence vector is a member of the set of x! Sequences; where x is the total number of city. After getting the initial population and associate initial sequence vector we calculate fitness value of each individual, according to equation (3). In the next step these individuals are visited by employed bees and onlooker bees of Bee colony. Employed and onlooker bee operator used for local search to avoid local max problem. In employed and onlooker bee phase new vector is generated and using SPV rule new sequence vector is also generated. Then the cost of this offspring is calculated. Using the sequence and its cost from the cost matrix the fitness value of the each individual is calculated. Algorithm 1 is the proposed algorithm ABC with SPV for the TSP.

**Algorithm 1: Travelling Salesman Problem using ABC with SPV rule**

```
[Initialization Phase]
for p=0 to population size do
    ford=0 to  dimension size do
        Randomly initialize particle
      Using SPV rule a sequence vector is generated
    end for d
Compute fitness of that particle
end for s
Repeat
[Employed Bee Phase]
```

```
        fori=0 to max no of employed bee do
            for d= 0 to dimension do
                produce new candidate solution
                produce candidate sequence on the basis of
                candidate solution using SPV rule
            end for d
            Compute fitness of individual
            if fitness of new candidate solution is better than
            the existing solution replace the older solution and
            its sequence vector.
        end for i
            Calculate the probability for each individual.
    [Onlooker Bee Phase]
        fori=0 to max no of onlooker bee do
            choose food source on the basis of probability
            for d= 0 to dimension do
                produce new candidate solution
                produce candidate sequence on the basis of
                candidate solution using SPV rule
            end for d
            compute fitness of individual
            if fitness of new candidate solution is better than
            the existing solution replace the older solution and
            its sequence vector.
        end for i
    [Scout Bee Phase]
         If any food source exhausted
then replace it by random position generated by scout
memorize the best solution and sequence found so far
      until(stopping criteria is not met)
```

## V. EXPERIMENTAL SETUP AND RESULTS

### A. Experiential Setup

For every algorithm there are some control parameters which are used for its efficient working. Hence, there are some control parameters for artificial bee colony algorithm with SPV rule also. We did an extensive literature survey and carried out our own experiments for determining the values of these control parameters. From this we found that the values which we have taken in this experiment are standard values and they are also suitable for this experiment.

The first control Parameter is Maximum cycle number: Maximum number of cycles (MCN) equals to the maximum number of generation, we have taken the result for 1500, 2000 and 2500 MCN value. The next parameter in our experiment is maximum number of population and we have taken its value to be 30 and 60.Another control parameter is number of runs and we have taken its value in our experiment as 30.It must be noted that each run contains maximum cycle number, which is 1500, 2000, 2500 in our experiment. The fourth control parameter is Dimension and it depends upon the number of city.

### B. Experimental Result

In this section we analyze the result obtained by our algorithm. To test the efficiency of our algorithm results of ABC with SPV is compared with real coded Genetic algorithm results. In a TSP we already have the information about the number of cities, path, and the cost associated with path that will be taken by a salesman to complete task. We just need to find the sequence of path which will provide us the optimal results. We conducted the experiment by varying the number of cities as well as varying the cost associated with path and then we compared our results with that of GA. In particular, we have taken two cases in which we have taken different number of cities and cost.

**Experiment 1:** Here, we are assuming there are 30 cities. Following are the execution time (in units) taken by GA and ABC with SPV.

**TABLE I**
**DISTANCE CALCULATED BY RBCA AND GA FOR 30 CITIES**

| No.of City | Cycles per evaluation | Means of 30 runs ABC | Means of 30 runs GA |
|---|---|---|---|
| 30 | 1500 | 335.93 | 981.8 |
| 30 | 2000 | 335.5 | 966.63333 |
| 30 | 2500 | 322.43 | 950.5 |

The sequence generated by GA is: 0,25,28, 23,14,18,19,22,16,9,5,6,24,15,4,11,13,21,12,27,1,20,2, 7,3,17,29, and 10,26,8.

The sequence generated by ABC with SPV is: 25, 12, 18, 14, 11, 29, 1, 27, 20, 16, 13, 24, 28, 3, 7, 8, 23, 17, 5, 6, 10, 0, 22, 19, 21, 15, 4, 2, 26, and 9.

**Experiment 2:** Here, we are assuming there are 60 cities. Following are the execution time (in units) taken by GA and ABC with SPV.

**TABLE II**
**DISTANCE CALCULATED BY RGBCA AND GA FOR 60 CITIES**

| No.of City | Cycles per evaluation | Means of 30 runs ABC | Means of 30 runs GA |
|---|---|---|---|
| 60 | 1500 | 825.767 | 2294.2 |
| 60 | 2000 | 798.566 | 2328.366667 |
| 60 | 2500 | 794.766 | 2294.2 |

The sequence generated by GA is: 0,35,34,17,9,51,59, 58, 19, 55, 13, 14, 33, 20, 39, 8, 56, 57, 45, 44, 2, 58, 12, 53, 25, 48, 38, 43, 21, 5, 4, 46, 6, 10, 36, 41, 10, 15, 18, 49, 32, 24, 50, 22, 52, 54, 11, 29, 47, 23, 56, 40, 27, 30, 42, 31, 7, 37, 1, 3.

The sequence generated by ABC with SPVis:30, 52, 9, 50, 20, 13, 48, 25, 41, 29, 33, 55, 54, 12, 32, 26, 46, 43, 22, 0, 15, 38, 36, 56, 19, 44, 5, 7, 49, 8, 14, 35, 28, 18, 51, 6, 42, 23, 37, 4, 1, 58, 39, 27, 10, 59, 16, 47, 3, 17, 31, 34, 53, 40, 11, 24, 21, 57, 45, 2.

The sequences generated by the algorithms shows the path for salesman to travel. The sequence shows the city number through which salesman travel one by one. For e.g. in sequence:25, 12, 18, 14, 11, 29, 1, 27, 20, 16, 13, 24, 28, 3, 7, 8, 23, 17, 5, 6, 10, 0, 22, 19, 21, 15, 4, 2, 26, and 9.

Salesman starts from city 25 than move on to city 12 and then city 18 and so on it last visits the city 9 and goes back to 25. The above sequence is generated by ABC with SPV for 30 cities.

From the Table I and II it is clear that our proposed ABC with SPV rule performs better than the real coded genetic algorithm in every case.

## VI. CONCLUSION

It can be concluded from the results TABLE I and II that proposed ABC with SPV performs better than the existing GA algorithm. There is no specific value for crossover and mutation probability used in genetic algorithm for which we can obtain best results for TSP. It depends upon number of cities and path. The procedure followed in TSP consists of the generation of the population according to the algorithm, then the path sequence and cost associated with the path sequence are generated, each individual and the path sequence vector set are updated using employed and onlooker operators. It is repeated again and again till the maximum number of cycles. As future work we have the intention to apply other types of nature inspired algorithms to the Traveling salesman problem, comparing their results with the ones accomplished by the ABC with SPV Algorithm. We can also use our proposed algorithm ABC with SPV rule to solve various optimization problems.

## REFERENCES

1. Applegate, D. L.; Bixby, R. M.; Chvátal, V.; Cook, W. J. (2006), “The Traveling Salesman Problem”, ISBN 0691129932.
2. C.A. Silvaa, lM.C. Sousaa, T.A. Runkler, "Rescheduling and optimization of logistic processes using GA and ACO," Engineering Applications of Artificial Intelligence., vol. 21, pp. 343-352, 2008.
3. M. Griltschel, M. Padberg, "Ulysses 2000: In Search of Optimal Solutions to Hard Combinatorial Problems," Technical Report, NewYork University Stern School of Business, 1993.
4. G. Zhao, W. Luo, H. Nie, C. Li, "A Genetic Algorithm Balancing Exploration and Exploitation for the Travelling Salesman Problem,"in Proceedings of the 2008 Fourth International Conference onNatural Computation, 2008, pp. 505-509.
5. Dervis Karaboga · Bahriye Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm” J Glob Optim (2007), pp 459-471
6. W.-L. Zhong, l Zhang, W.-N. Chen, "A Novel Discrete Particle Swarm Optimization to Solve Traveling Salesman Problem," in Proc. IEEE Int. Conf. Evol. Comput. (CEC), 2007, pp. 3283-3287.
7. L.-P. Wong, M.Y. Hean Low, C.S. Chong, "A Bee Colony Optimization Algorithm for Traveling Salesman Problem," Second Asia International Conference on Modelling & Simulation, 2008,pp. 818-823.
8. S. Nonsiri, S. Supratid, "ModifYing Ant Colony Optimization," IEEE Conference on Soft Computing in Industrial Applications, 2008,pp. 95-100.
9. Marco Dorigo. “Ant Colonies for the Traveling Salesman Problem”. IRIDIA, UniversitéLibre de Bruxelles. IEEE Transactions on Evolutionary Computation, 1(1):53–66. 1997.
10. D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey, Technical Report-TR06, 2005.
11. D. Karaboga, B. Akay, "A Survey: Algorithms Simulating Bee Swarm Intelligence," Artificial Intelligence Review., vol. 31, pp. 68-85, 2009.
12. D. E. Goldberg, “Genetic Algorithm in Search. Optimization and Machine Lerning”, Addison Wesley, 1989.
13. G-. G. Jin, S-. R. Joo, “A Study on a Real-Coded Genetic Algorithm,”
14. Journal of Control, Automation, and Systems Engineering, vol. 6, no. 4, pp. 268-274, April 2000.
15. H. Sengoku and I. Yoshihara, "A Fast TSP Solver using GA on JAVA," in Proc. of 3rd International Symposium on Artificial Life and Robotics, vol. I, 1998, pp. 283-288.
16. G.A. Croes, "A method for solving traveling salesman problems,"
17. Operations Research.vol. 6, pp. 791-812, 1958.
18. G. Reineit, "TSPLIB-A traveling salesman problem library," ORSA Journal on Computing., vol. 3, no. 4, pp. 376-384, 1991.
19. XH. Shi, Y.e. Liang, H.P. Lee, C. Lu, Q.x. Wang, "Particle swarm optimization-based algorithms for TSP and generalized TSP," Information Processing Letters., vol 103, pp. 169-176,2007.
20. L. Davis, “Handbook of Genetic Algorithms”, Von Nostrand Reinhold, N.Y., 1991