Volume 66, Issue 4, 2022

Journal of Scientific Research

of

The Banaras Hindu University

Solving Travelling Salesman Problem using Artificial Immune System Optimization (AISO)

Ranjit Kr Mandal*1, Pinaki Mukherjee2, and Mousumi Maitra3

*1Dept. of Computer Sc. & Engineering, Govt. College of Engineering & Ceramic Technology, Kolkata, India, ranjitgcect@gmail.com
2Dept. of ECE, Govt. College of Engineering & Ceramic Technology, Kolkata, India, pinakimukherjeer@gcect.ac.in
3Deptof IT, Govt. College of Engineering & Ceramic Technology, Kolkata, India, mou1232005@.yahoo.com

Abstract: **Travelling Salesman Problem (TSP) is a typical NP complete combinatorial optimization problem with various applications. In this paper, a nature inspired meta-heuristic optimization algorithm named as Artificial Immune System Optimization (AISO) algorithm is proposed for solving TSP. There are other approaches for solving this problem, namely Greedy Method, Brunch and Bound (B&B), and Dynamic Programming (DP) but they are not very efficient. The time complexity of Greedy approach is $O(n^2)$. However, the Greedy method doesn't always converge to an optimum solution whereas the B&B increases search space exponentially and DP finds out optimal solution in $O(n^2 2^n)$ time. The population based meta-heuristic optimization algorithms such as Artificial Immune System Optimization (AISO) and Genetic Algorithm (GA) provide a way to find solution of the TSP in linear time complexity. The proposed algorithm finds out the best cell (optimum solution) using a Survivor Selection (SS) operator which reduces the search space to ensure that effective information is not lost. Dataset, results and convergence graphs are presented and accuracy of the analysis is briefly discussed.**

Index Terms: **Artificial Immune System Optimization (AISO), Dynamic programming (DP), Genetic Algorithm (GA), SS Operator, Traveling Salesman Problem (TSP).**

I. Introduction

The Travelling Salesman Problem (TSP) is a well known typical NP complete combinatorial optimization problem to find the minimum distance tour of a salesman who starts from his home city C_1 and covers all n cities exactly once and coming back to his home city C_1 (Schrijver,1960). TSP has numerous applications in computer wiring, vehicle routing (Lenstra & Rinnooy,1975), drilling problem of printed circuit boards (PCBs) (Grötschel et al., 1991), overhauling gas turbine engines (Plante et al., 1987) and X-Ray crystallography (Bland & Shallcross, 1989). TSPs are classified into symmetric TSP, asymmetric TSP, and multi TSP (Rajesh et al., 2010). Different approaches have been proposed to solve TSP, which can be classified into two categories: deterministic algorithms andmeta-heuristic algorithms.

Deterministic algorithms do not involve any randomness in the model. But it is a rigorous procedure. Greedy method (Sk. Mastan et al., 2019), Branch and bound algorithm (B&B) (Saad et al., 2013) and dynamic programming (R. Bellman, 1966), (V. B. Lobo et al., 1916) are the typical deterministic algorithms for solving TSP. Deterministic algorithms perform well on the TSP of small number of city tour. However, with the increase of the number of cities, greedy method does not produce optimal solution and the search space increases exponentially for B&B. The performance of deterministic algorithms degrade significantly. So, deterministic algorithms are not suitable for optimizing the TSP of large number of city tour.

Meta-heuristic optimization algorithm is a kind of stochastic algorithm which can accelerate the optimization process and find solutions in reasonable time but not guaranteeing to find the optimal solution.

Genetic Algorithm (GA) (J. McCall, 2005) and Ant Colony Optimization (ACO) (Dorigo& Gambardella, 1996; Dorigo & Gambardella, 1997) are some of the population based Meta-heuristic optimization algorithms which are successfully applied for solving TSP.

Artificial Immune System (AIS) is a population based meta-heuristic optimization algorithm which is inspired by structure, functions, models and information processing mechanism of biological immune system. Artificial Immune Systems and their applications are introduced by D. Dasgupta (Dasgupta, 1999).

* Corresponding Author

In this paper, AIS is proposed for solving TSP. The main contribution of this paper is illustrated as follows:

Firstly, a Generate Function which randomly generates the initial population (feasible solutions) is developed.

Secondly, AIS is implemented for solving TSP.

Finally, the performance of the proposed AIS is compared to several state-of-art TSP algorithms.

II. Travelling Salesman Problem (TSP)

The TSP is a well known typical NP complete combinatorial optimization problem to find the minimum distance tour of a salesman who starts from his home city c_1 and covers a given set (N) of customer cities and coming back to his home city c_1.

The weighted graph,

$$G = (N, A)$$

N = set of nodes representing the cities.

A = set of fully connected arcs of N nodes.

Each arc assigned a value d_{ij}, which is the distance of the arc (i, j) Є A in the distance between cities i, j with (i, j) Є N. Fig 1 shows a typical example graph with 5 cities and the corresponding distance matrix.

The TSP is the problem of finding a minimal distance Hamiltonian cycle of the graph, where a Hamiltonian cycle is a closed tour visiting exactly once each of the n =N nodes of G.

For n cities to visit, let X_{ij} has the variable that has value 1 if the salesman goes from city i to city j and the value 0 if the salesman does not go from city i to city j. Let d_{ij} is the distance from city i to city j.

Mathematically, we can define TSP (Lenstra & Rinnooy, 1975) as follows:

$$Minimize\ \sum_{i=1}^{n}\sum_{j=1}^{n} X_{ij}\, d_{ij} \qquad (1)$$

Subject to

$$\sum_{i=1, i\neq j}^{n} X_{ij} = 1 \text{ for j=1,2,3,...n} \qquad (2)$$

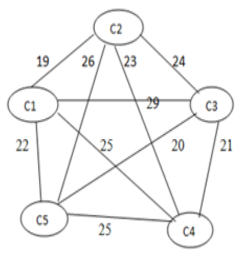
$$\sum_{j=1, i\neq j}^{n} X_{ij} = 1 \text{ for i=1,2,3,...n} \qquad (3)$$

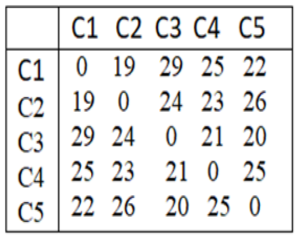
$$\sum X_{ij} = 0 \quad \text{for all i and j} \qquad (4)$$

If the d_{ij}= d_{ji} for all (i, j), TSP is called symmetric, otherwise it is called asymmetric.

III. Artificial Immune System (AIS)

Artificial Immune System (AIS) is a population based meta-heuristic optimization algorithm which is mimicking the structure, functions, models and information processing mechanism of biological immune system (Dasgupta, 1999). The AIS begin with a set of cell (solution) known as population. According to fitness value total population are classified into two categories such as molecule and antigen. After that new cells are formed which are copies of their parents (clone). New offspring is produced by muting cloned molecule and antigen using two points crossover concept.





	C1	C2	C3	C4	C5
C1	0	19	29	25	22
C2	19	0	24	23	26
C3	29	24	0	21	20
C4	25	23	21	0	25
C5	22	26	20	25	0

Fig. 1. An Example graph and its adjacency matrix representation

The main steps involved in AIS are presented below:

Step 1: Initialize Population: Population is to be initialized.

Step 2: Fitness calculation: The fitness value for each cell is to be calculated.

Step 3: Sorting: Cells are sorted in ascending order according to fitness.

Step 4: Cloning: New cells are formed which are copies of their parents.

Step 5: Mutation: Mutation is done between molecule (good fitness) and antigen. This operation produces new offspring.

Step 6: Combine parent and offspring.

Step 7: Survivors Selection: Best individuals are selected for next generation.

Step 8: Check whether the stopping condition is satisfied. If yes, stop. If no, go to Step 4.

IV. Artificial Immune System Optimization (AISO) for solving TSP

Although many researchers have been proposed many AIS algorithm for continuous optimization problem in recent years, the study of discrete AIS algorithm is fewer, which is the motivation of this paper. In order to solve TSP, an AISO algorithm is proposed, implemented and analyzed in this paper. The pseudo code of AISO algorithm is represented in Algorithm 1. Flowchart of AISO is represented in Fig. 2.

Algorithm1. AISO for solving TSP

```
Initialization
1.   Set the value of the population size (N), molecule selection factor (α), the distance matrix of city tour and maximum number of generation MGN

Task 1: Initial population generation and evaluation fitness value
2.   $for\ i = 1\ to\ N\ do$
3.   Generate an initial population Xi randomly as Eq. (2) and Eq. (3)
4.   Evaluate the fitness value of $f$(Xi) as Eq. (5)
5.   $end\ for$
6.   Sort N individuals in ascending order based on fitness value and store as memory

Task 2: Cloning, Mutation and SS operation
7.   $while\ t < MGN\ do$
8.   for each individual generate a clone
9.   Store first (α x N) clones as molecule and rest of the clones as antigen
10.  $for\ i = 1\ to\ N\ do$ (Population size)
11.      Select a molecule and an antigen randomly
12.  Generate antibody by mutation operation between selected molecule and antigen (using two points crossover)
13.  Evaluate fitness value of $f$(antibody) as Eq. (5)
14.  $end\ for$
15.  Combined parent population and antibodies
16.  Select N best individual as population for next generation
17.  t=t+1
18.  $end\ while$
19.  Return best solution
20.  $end$
```

The main components of AISO are presented as following.

A. Input function

In this phase, population size (*N*), maximum number of generation (MGN) and *k* x *k distance (d)* matrix of city tour are given. Hence, *d(i, j)* is the distance from city c_i to city c_j. If there is no direct path from city c_i to city c_j, *d(i, j)* is assumed a large value.

B. Generating Function

Generating function of proposed AISO generates *N* cells using 1 to *k* (city) digit as Eq. (2) and Eq. (3). Cell is one-dimension array. Size of array is equal to the number of city of the TSP. For example, the following cell is generated for the above TSP in Fig 1. of section II:

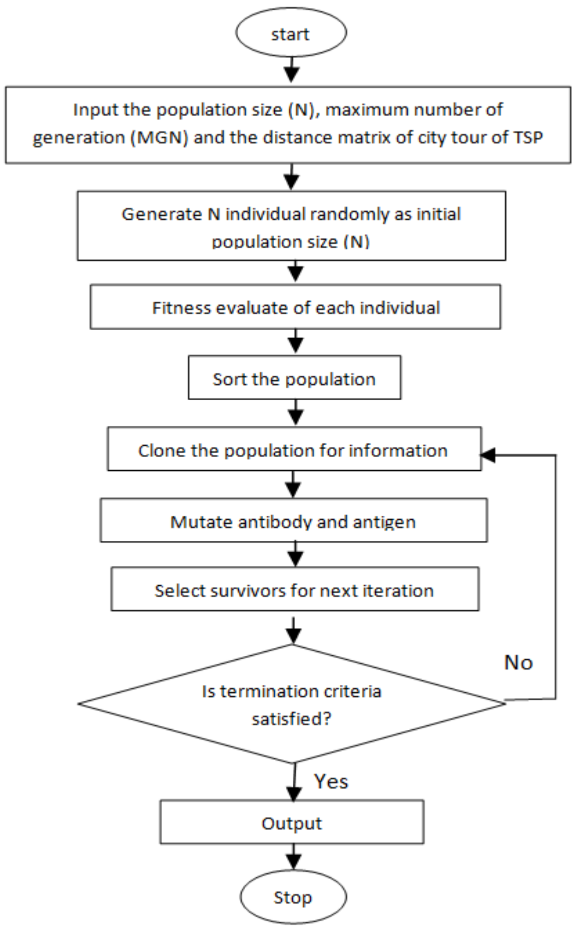

start
Input the population size (N), maximum number of generation (MGN) and the distance matrix of city tour of TSP
Generate N individual randomly as initial population size (N)
Fitness evaluate of each individual
Sort the population
Clone the population for information
Mutate antibody and antigen
Select survivors for next iteration
Is termination criteria satisfied?
No
Yes
Output
Stop

Fig. 2 Flowchart of the AISO

1	2	3	4	5

C. Fitness Calculator

Fitness calculator of our proposed AISO computes the fitness value of each cell of the population by Eq. (5) as follows

$$\text{Fitness} = \sum_{i=1}^{k-1} d(i, i+1) + d(k, 1) \quad (5)$$

For example, the calculated fitness value using Eq. (5) of individual cell is shown in 3rd column of the Table I

Table I. Initial population and their fitness value

String No.	Initial Population	Fitness value
1	5 1 3 4 2	121
2	2 3 1 4 5	129
3	4 3 5 1 2	105
4	5 1 3 2 4	123
Total Fitness value		478
Average Fitness value		119.5
Best Fitness value		105

D. *Sorting Operator*

Sorting operator of proposed AISO sorts *N* number of cells in ascending order according their fitness value and stores.

E. *Cloning Operator*

Cloning operator of our proposed AISO generates new cells by copying their parent cells for keeping information.

F. *Mutation Operator*

Mutation operator of proposed AISO firstly partitions the population into two categories such as α molecule and (*N*-α) antigen. Hence we assume α is 0.5* *N*. Then select one cell form molecule and one cell from antigen randomly. New offspring antibody cells are produced by muting selected molecule cell and antigen cell using two points crossover concept.

G. *Survivors Selection (SS) Operator*

SS operator of our AISO selects best *N* cells based on fitness value among parent population and antibodies for next iteration. Thus, it controls the population, reduces searching space and increases efficiency of AISO. For an example, four best cells are selected for next generation by SS operator of our AISO as shown in Table II.

Table II. Selected cells for next iteration

Cell No.	Selected Cells for next iteration	Fitness value
1	4 3 5 1 2	105
2	4 1 2 3 5	113
3	4 1 2 3 5	113
4	5 3 1 2 4	116

H. *Termination Operator*

It checks whether the end condition is satisfied. If not, repeats from (E) Fitness operator to (G) SS operator.

I. *Output operation*

This phase of AISO gives optimal solution and stop.

V. Results and Discussion

This section presents dataset details, results obtained and discussion. The overall performance of the proposed AISO is examined by using benchmark instances of the TSP. The instances of TSP are taken from various sources. The success rate is calculated by the number of times optimal result obtained divided by the number of times the program is executed on a TSP

The algorithm is coded in Python 3.9 with Pycharm IED and experiments are carried out using a Intel Core(TM) i7-5500U CPU @2.40GHz, 8GB RAM and operating system is Windows 10.

Table III. Parameter Configuration

Parameters of AISO
Population size: up to 100
Maximum Iteration (MGN): 300
Molecule selection factor (α) : 0.5 * N

The propose AISO is illustrated in the following example

Example 1 A graph structure is modeled from the airline network of Indigo airlines as Fig 4 by using the air distances between the cities to find optimal distance tour of a salesman who starts from his home city Delhi and covers all cities exactly once and coming back to his home city Delhi (Nagoor et al., 2016).

Fig. 3 Route map of Indigo airlines

From the above map in Fig 3, flight routes between the cities Delhi, Mumbai, Bangalore, Chennai, Coimbatore, Kochi and Trivandrum are selected. Fig 4 represents the graph of the airline route map in which the vertices denotes the cities Delhi (1), Mumbai (2), Chennai (3), Bangalore (4), Kochi (5), Coimbatore (6) and Trivandrum (7) and the edge weight is the air distance (K.M.) between two cities.

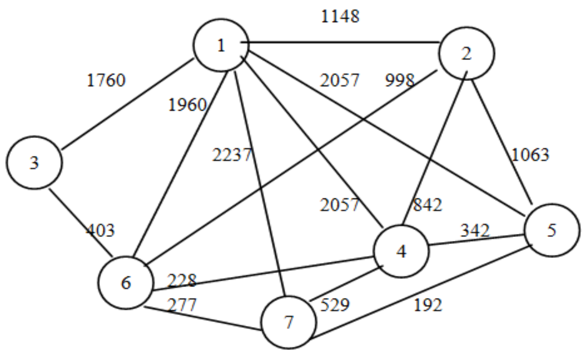


Fig. 4 Graph of selected cities for TSP instance

Fig 5 represents obtained minimal distance Hamiltonian cycle of AISO and GA for TSP instance of Fig 4. The total distance of the tour as shown in Fig 5 is (1760 + 403 +277 + 192 + 342 +842 + 1148) 4964 K.M. The success rate is 100 %

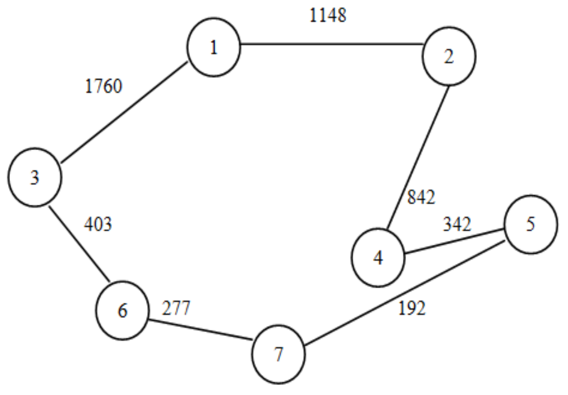


Fig. 5 Obtained optimal tour for above TSP instance

Dataset: This dataset used for different algorithms to solve the TSP instance was obtained from stack overflow.com/questions/11007355/data-for-simple-tsp (Data_for_simple_tsp) and people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html (Datasets) for the TSP problem as shown in Table IV and Table V. Number of city, optimal distance and distance matrix are given in the Table IV and distance matrix of 15 cities are given in Table V

Table IV. Number of city, optimal distance and distance matrix of TSP data set

TSP Ins.	Cities	Optimal distance	Distance Matrix
TSP1	4	35	0 10 15 20 5 0 9 10 6 13 0 12 8 8 9 0
TSP2	5	28	0 20 30 10 11 15 0 16 4 2 3 5 0 2 4 19 6 18 0 3 16 4 7 16 0
TSP3	5	105	0 19 29 25 22 19 0 24 23 26 29 24 0 21 20 25 23 21 0 25 22 26 20 25 0
TSP4	5	19	0.0 3.0 4.0 2.0 7.0 3.0 0.0 4.0 6.0 3.0 4.0 4.0 0.0 5.0 8.0 2.0 6.0 5.0 0.0 6.0 7.0 3.0 8.0 6.0 0.0
TSP5	6	1248	0 64 378 519 434 200 64 0 318 455 375 164 378 318 0 170 265 344 519 455 170 0 223 428 434 375 265 223 0 273 200 164 344 428 273 0
TSP6	11	253	0 29 20 21 16 31 100 12 4 31 18 29 0 15 29 28 40 72 21 29 41 12 20 15 0 15 14 25 81 9 23 27 13 21 29 15 0 4 12 92 12 25 13 25 16 28 14 4 0 16 94 9 20 16 22 31 40 25 12 16 0 95 24 36 3 37 100 72 81 92 94 95 0 90 101 99 84 12 21 9 12 9 24 90 0 15 25 13 4 29 23 25 20 36 101 15 0 35 18 31 41 27 13 16 3 99 25 35 0 38 18 12 13 25 22 37 84 13 18 38 0

Table V. Distance matrix of 15 cities TSP data set

TSP Ins.	Distance Matrix

TSP7	-1 141 134 152 173 289 326 329 285 401 388 366 343 305 276
	141 -1 152 150 153 312 354 313 249 324 300 272 247 201 176
	134 152 -1 24 48 168 210 197 153 280 272 257 237 210 181
	152 150 24 -1 24 163 206 182 133 257 248 233 214 187 158
	173 153 48 24 -1 160 203 167 114 234 225 210 190 165 137
	289 312 168 163 160 -1 43 90 124 250 264 270 264 267 249
	326 354 210 206 203 43 -1 108 157 271 290 299 295 303 287
	329 313 197 182 167 90 108 -1 70 164 183 195 194 210 201
	285 249 153 133 114 124 157 70 -1 141 147 148 140 147 134
	401 324 280 257 234 250 271 164 141 -1 36 67 88 134 150
	388 300 272 248 225 264 290 183 147 36 -1 33 57 104 124
	366 272 257 233 210 270 299 195 148 67 33 -1 26 73 96
	343 247 237 214 190 264 295 194 140 88 57 26 -1 48 71
	305 201 210 187 165 267 303 210 147 134 104 73 48 -1 30
	276 176 181 158 137 249 287 201 134 150 124 96 71 30 -1

TSP8(p01.tsp)	0 29 82 46 68 52 72 42 51 55 29 74 23 72 46
	29 0 55 46 42 43 43 23 23 31 41 51 11 52 21
	82 55 0 68 46 55 23 43 41 29 79 21 64 31 51
	46 46 68 0 82 15 72 31 62 42 21 51 51 43 64
	68 42 46 82 0 74 23 52 21 46 82 58 46 65 23
	52 43 55 15 74 0 61 23 55 31 33 37 51 29 59
	72 43 23 72 23 61 0 42 23 31 77 37 51 46 33
	42 23 43 31 52 23 42 0 33 15 37 33 33 31 37
	51 23 41 62 21 55 23 33 0 29 62 46 29 51 11
	55 31 29 42 46 31 31 15 29 0 51 21 41 23 37
	29 41 79 21 82 33 77 37 62 51 0 65 42 59 61
	74 51 21 51 58 37 37 33 46 21 65 0 61 11 55
	23 11 64 51 46 51 51 33 29 41 42 61 0 62 23
	72 52 31 43 65 29 46 31 51 23 59 11 62 0 59
	46 21 51 64 23 59 33 37 11 37 61 55 23 59 0

For every data instance of dataset, AISO and GA ware executed at least 20 times. The optimal distance and obtained solutions of AISO and GA are presented in Table VI. It is seen that success rate of AISO on TSP 7 instance is not so good.

Table VI.Result obtained on TSP data set

TSP Ins.	Algorithms	Obtained optimal distance	optimal path	Success Rate (%)
Example 1	AISO	4964	1-3-6-7-5-4-2-1	100
	GA	4964	1-3-6-7-5-4-2-1	100
TSP1	AISO	35	1-2-4-3-1	100
	GA	35	1-2-4-3-1	100
TSP2	AISO	28	1-4-2-5-3-1	100
	GA	28	1-4-2-5-3-1	100
TSP3	AISO	105	1-2-4-3-5-1	100
	GA	105	1-2-4-3-5-1	100
TSP4	AISO	19	1-4-5-2-3-1	100
	GA	19	1-4-5-2-3-1	100
TSP5	AISO	1248	1-2-3-4-5-6-1	100
	GA	1248	1-2-3-4-5-6-1	100
TSP6	AISO	253	1-8-5-4-10-6-3-7-2-11-9-1	90
	GA	253	1-8-5-4-10-6-3-7-2-11-9-1	90
TSP7	AISO	1194	1-2-15-14-13-12-11-10-9-8-7-6-5-4-3-1	50
	GA	1194	1-2-15-14-13-12-11-10-9-8-7-6-5-4-3-1	100
TSP8	AISO	291	1-11-4-6-8-10-14-12-3-7-5-9-15-2-13-1	70
	GA	291	1-11-4-6-8-10-14-12-3-7-5-9-15-2-13-1	100

Fig. 6, Fig. 7 and Fig. 8 represent comparison graph for the performance of AIS and GA on TSP6, TSP7 and TSP8 respectively. X axis and Y axis represent Average fitness and number of iteration respectively. It can be seen from the convergence graph that the performance of AISO is better than Genetic Algorithm.

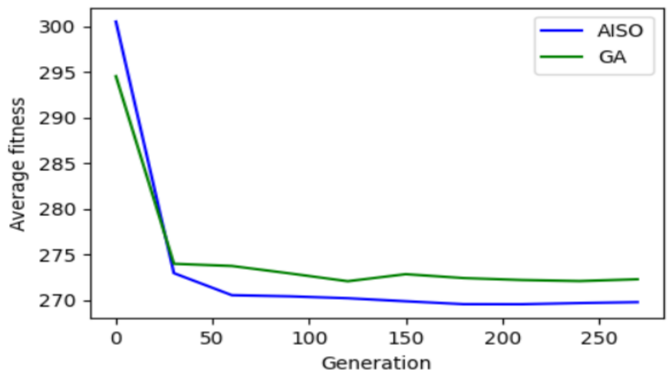


Fig. 6 Convergence graph of TSP6

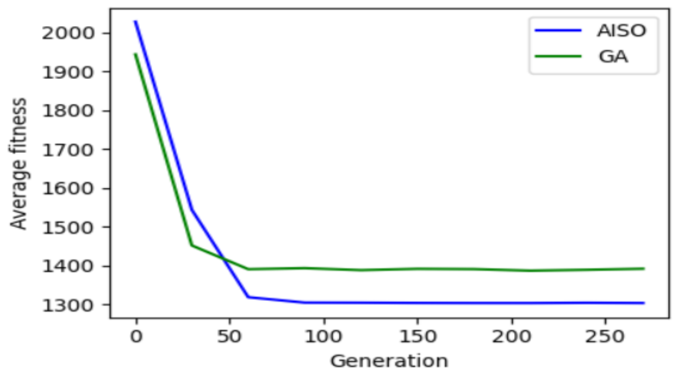


Fig. 7 Convergence graph of TSP7

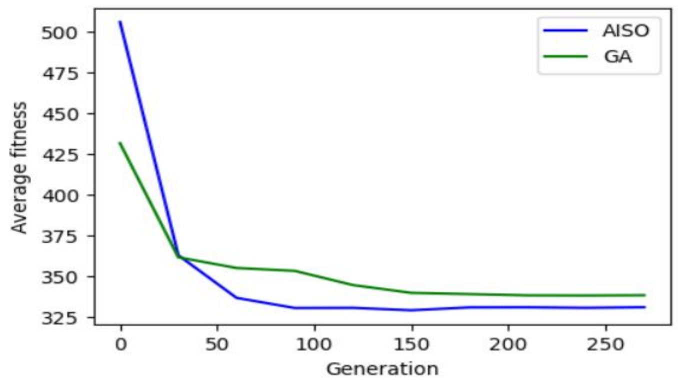


Fig. 8 Convergence graph of TSP8

Conclusion

In this paper, a novel meta-heuristic discrete Artificial Immune System Optimization (AISO) algorithm is proposed for solving TSP. The key point of the AISO is implementing a mutation operator which generates antibody for surviving. In order to evaluate the performance of AISO one TSP dataset is employed. The experimental results demonstrate that the proposed AISO is highly efficient algorithm in respect of computational complexity, accuracy, convergence capability, and stability.
For future studies, there is a scope to improve success rate AISO on big TSP instance. In future, we will apply AISO to solve other NP complete optimization problems, such as Quadratic Assignment Problem.

Acknowledgment

Authors are thankful to the "Center Computing Lab", "Project on a population-basedmeta-heuristicalgorithm for optimization" of Computer Science & Engineering Department, Government College of Engineering and CeramicTechnology, Kolkata, for providing infrastructural facilities during progress of the work. Authors, Mr. Ranjit Kr Mandal, Dr. Pinaki Mukherjee and Mousumi Maitra are thankful to Government College of

Engineering and CeramicTechnology, Kolkata for kindly permitting them to carry on the research work.

.

References

Bland, R.E., & D.E Shallcross,(1989).Large traveling salesman problem arising from experiments in X-ray crystallography: a preliminary report on computation. Operations Research Letters, Vol. 8(3), pp. 125-128,

D.Dasgupta,(1999) Artificial Immune Systems and their Applications ,Springer- Verlag,

Data_for_simple_tsp.https://stackoverflow.com/questions/11007355/data-for-simple-tsp

Datasets. https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html

Dorigo, M. & Gambardella, L.M.(1996). “Ant Colonies for the Traveling Salesman Problem”, University Libre de Bruxelles, Belgium.

Dorigo, M. & Gambardella, L.M. (1997). Ant Colony System: A Cooperative Learning Application to Traveling Salesman Problem, IEEE Transactions on Evolutionary Computation, Vol.1, No.1.

Grötschel, M.; M. Jünger& G. Reinelt.(1991). Optimal Control of Plotting and Drilling Machines: A Case Study. Mathematical Methods of Operations Research, Vol. 35, No. 1, pp.61-84.

J. McCall,(2005). ‘‘Genetic algorithms for modelling and optimisation,’’ J. Comput. Appl. Math., vol. 184, no. 1, pp. 205–222

Johnson D.S. &McGeoch L.A.(1995). The Traveling Salesman Problem: A Case Study in Local Optimization.

NagoorGani, A., Latha, S.R, (2016). A new algorithm to find fuzzy Hamilton cycle in a fuzzy network using adjacency matrix and minimum vertex degree. *SpringerPlus* **5,** 1854 https://doi.org/10.1186/s40064-016-3473-x

Lenstra, J.K,.&RinnooyKan,(1975). A.H.G.Some simple applications of the traveling salesman problem. Operational Research Quarterly, Vol. 26, pp. 717–33,

Plante, R.D.; T.J. Lowe & R. Chandrasekaran.(1987). The Product Matrix Traveling Salesman Problem: An Application and Solution Heuristics. Operations Research, Vol. 35, pp. 772-783,

R. Bellman,(1966). ‘‘Dynamic programming,’’ Science, vol. 153, nos. 37–31, pp. 34–37,

Rajesh Matai, Surya Singh &Murari Lal Mittal,(2010). Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches | InTechOpen, doi:10.5772/12909

Saad, Shakila& Wan Jaafar, Wan Nurhadani& Jamil, Siti, (2013). Solving Standard Traveling Salesman Problem and Multiple Traveling Salesman Problem by Using Branch-and-Bound. AIP Conference Proceedings. 1522. 1406-1411. 10.1063/1.4801294,

Schrijver, A., (1960). On the history of combinatorioal optimization,

Sk. Mastan, U. Balakrishna& G. SankarSekhar Raju., (2019). Heuristics Designed For the Traveling Salesman Problem, International Journal of Research in Advent Technology, Vol.7, No.5.

V. B. Lobo, B. B. Alengadan, S. Siddiqui, A. Minu and N. Ansari, (2016). "Traveling Salesman Problem for a Bidirectional Graph Using Dynamic Programming," *International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, pp. 127-132, doi: 10.1109/ICMETE.2016.26

***