iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** VIII **Month of publication:** August 2017

DOI: http://doi.org/10.22214/ijraset.2017.8101

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VIII, August 2017- Available at www.ijraset.com

Design and Development of Hybrid Genetic Algorithm Based Ant Colony Optimization Approach for Routing

Harleen Kaur[1], Er. Haramandeep Singh[2]
[1]Student, [2]Assistant Professor, Department of Computer Engineering, Punjabi University, Patiala

Abstract: Travelling Salesman Problem (TSP) is the most classically solved problems in computation mathematics in recent years. It is a classical combinatorial optimization and NP-complete problem. TSP is a problem which targets to find a possible tour along which salesman has to visit each city exactly once from a given list of cities and must come back to the starting point of the tour so that total cost spent and distance covered is minimum. In this paper, a hybrid approach combining Genetic Algorithm (GA) with Ant Colony Optimization (ACO) is conferred to solve TSP. In the proposed hybrid GA-ACO algorithm, it is modelled with swap mutation technique and probabilistic selection technique with described update condition. It is enforced to datasets from the standard database of TSPLIB in comparison with the other algorithms in order to appraise the performance of the proposed algorithm. An experiment was carried out and the experimental results show that the method is correct. The empirical results illustrate that the proposed hybrid approach can deal with the TSP well, and the developed mutation process and crossover process are effective.
Keywords: Travelling Salesman Problem; Genetic Algorithm; Ant Colony Optimization; Crossover; Swap Mutation, Pheromone Trail.

I. INTRODUCTION

The most classically studied problem in computational mathematics is the Travelling Salesman Problem (mostly called TSP). It can be stated as follows. A salesman wants to visit each city exactly once and returns to the origin city, in a given pool of cities and the distances between each pair of cities. The main question arises here is how to find the shortest possible route. It is significantly necessary research area as in practical applications TSP is a special case of the travelling purchaser problem and the vehicle routing problem.With the increase in the number of cities, the worst-case running time for any algorithm increases exponentially because the TSP resides to the class of NP-complete problems. In late years, swarm intelligence algorithms have been developed for the TSPs. An ACO described by Dorigo and Gambardella for solving TSP by using information collected in the form of a pheromone road accumulated on the edges of the TSP graph (Dorigo & Gambardella, 1997). The generalized travelling salesman problem discussed by Yang et al. extended the ACO method to this field (Yang, Shi, Marchese, & Liang, 2008). Dong et al. proposed a hybrid algorithm, cooperative genetic ant system to deal with TSP (Dong, Guo, & Tickle, 2012). A genetic algorithm describes by Dwivedi et al. solves the TSP by special crossover technique, Sequential Constructive Crossover (SCX) that develops a high-quality solution to the TSP (Dwivedi, Chauhan, Saxena, & Agrawal, 2012). Roy et al. discussed an effective adaptive genetic algorithm to solve the constrained solid TSP in crispy, fuzzy and rough environments using probability selection technique and adaptive crossover operator modeled with random mutation (Roy, Maity, & Maiti, 2015). Bidirectional constructive crossover, an evolutionary approach discussed by Kang et al. to solve the TSP using special crossover operation called SCX with bidirectional and circular search in the construction of off-springs (Kang, Kim, Won, & Kang, 2015). Masutti and Castro proposed a bee-inspired algorithm to solve the TSP by taking optBees algorithm for continuous optimization (Masutti & Castro, 2016). Jiang discussed discrete bat algorithm for solving the TSP (Jiang, 2016). Mahi et al. proposed a hybrid method by using particle swarm optimization for detecting optimum values of ACO algorithm (Mahi, Baykan, & Kodaz, 2015). Zhou and Song discussed partheno-genetic algorithm to solve the TSP by using new selection and mutation operators (Zhou & Song, 2016).

Genetic Algorithm is a computational intelligence algorithm which is developed by search, crossover and mutation techniques. In this paper, a GA with ant colony optimization (ACO) is presented for TSP. The remainder of this paper is organized as follows. In Section II a brief introduction to GA is conferred. In Section III, a precise introduction is made to the hybrid GA and ACO. In

©IJRASET (UGC Approved Journal): All Rights are Reserved

Section IV, the proposed algorithm is tested on the TSPLIB, EIL51, and S70 in order to verify the performance and then compared the results with existing algorithm, and the conclusions in Section VI.

II. GENETIC ALGORITHM

A category of optimization algorithm that is used to find the optimal solution to a given computational problem that maximizes or minimizes a particular function is the genetic algorithms. GA is an optimization approach based on natural evolution (Dwivedi, Chauhan, Saxena, & Agrawal, 2012). Evolutionary computation is a branch of the field of study is represented by GAs in which they act like the biological processes of reproduction and natural selection to get the fittest solutions. As GAs are devised to reproduce a biological process, because of this much of the suitable terminology is hired from biology. The basic factors of GAs are:

A. fitness function;
B. a population of chromosomes,
C. selection of which chromosomes will reproduce,
D. crossover to produce next generation of chromosomes
E. and mutation of chromosomes in new generation randomly

The genetic algorithm needs fitness function as the basis, the fitness function is the function that the algorithm is trying to optimize (Zhou & Song, 2016). The word 'fitness' is used here because the fitness function tests and computes how 'fit' each potential solution is. The term chromosome assigned here is the numerical value or values that show a candidate solution to the problem that GA is trying to solve. It is often encoded as a bit string, array of parameter values and a process that is also found in other optimization algorithms.

III.GA-ACO HYBRIDIZATION FOR TSP

In this section, a hybrid algorithm is proposed by the combination of GA and ACO and making insertion idea from chromosome structure. Then the proposed algorithm is used for solving TSP.

A. Selection Operator

Selection is an elitism which means the good solution will remain in the population and discard the bad solutions. For reproduction, the selection operator selects the chromosomes from the population. The time it is likely to be selected to reproduce depends upon how fit the chromosome is. Probability selection is used here to select the fittest chromosome depending on the corresponding probabilistic values. To minimize the cost objective, it is better to choose that population which is in the neighborhood of the minimum solution to the entire solution space. From the initial population select the best-fitted solution for TSP.

B. Crossover Operator

Crossover is a process of combining multiple candidate solutions to get new solutions. There is a combination of two solutions to get two new solutions or combination of two solutions to get one new solution. In short, there is a combination of 'm' number of solutions to get 'n' number of new solutions. The crossover operator is modeled on chromosomes. The crossover operator used in the proposed algorithm selects a subset of the parent string and then add a substring as a child. Any missing value is then added to the child from the second parent in order that it is found. It could be said that the main categorizing feature of a GA is the use of crossover. The crossover is explained below.

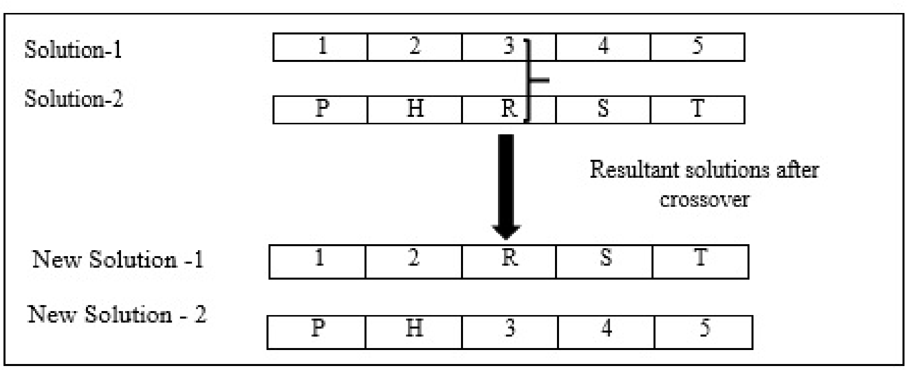


Figure 1 Crossover

©IJRASET (UGC Approved Journal): All Rights are Reserved

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VIII, August 2017- Available at www.ijraset.com

C. Mutation Operator

Mutation is a process which is capable to shuffle the route. It should ever add or remove a location from the root, otherwise, it would risk creating an invalid solution. In the proposed algorithm, swap mutation is used. Swap mutation is an effective mutation for TSP. With swap mutation, two locations are selected at random and their positions are a simple to swap. For eg., a mutation list of (P, H, **S**, B, **M**) might end with (P, H, **M**, B, **S**) because swap mutation is only swapping pre-existing value. It will never create a list of missing or duplicate value and that exactly we need for TSP.

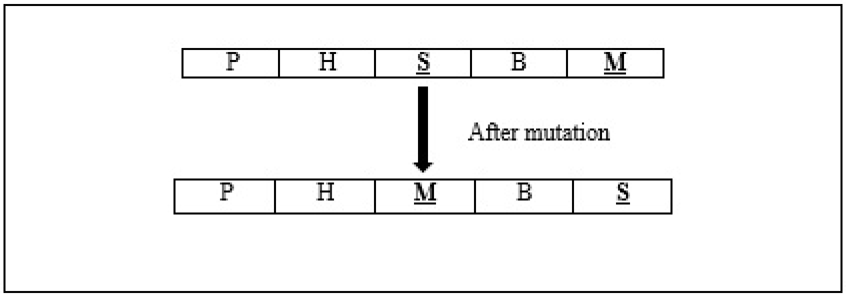


Figure 2 Mutation

IV. DESCRIPTION OF PROPOSED ALGORITHM

GA-ACO maintains the algorithmic simplicity of GA and ACO. The pseudo code of the proposed algorithm can be summarized in Algorithm 1. The terminal condition is the maximum number of fitness evaluations.

Algorithm 1: Pseudo code of GA-ACO
Begin;
Initialize the population randomly in a dimension space;
Calculate the fitness of all solutions in population;
while
if score ≠ good_enough
for GA-ACO
do
{
place ant cluster = random city;
while
all city ≠ visited;
prob_q = best city selected;

$P_{ij} = \frac{(\tau_{ij})^{\alpha}(n_{ij})^{\beta}}{\sum_{k \in allowed\ tasks}(\tau_{ik})^{\alpha}(n_{ik})^{\beta}}$, --------(Eq. 1)

$if\ j \nexists tabu\ list;$

$else\ 0;$
}
Update condition
{
for ij in the edge set
if edge ij = traversed;

$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \Delta\tau$; --------(Eq. 2)

Else

$\tau_{ij} = (1-\rho)\,\tau_{ij};$ --------(Eq. 3)

}
Update;
End;

©IJRASET (UGC Approved Journal): All Rights are Reserved

where $(\tau_{ij})^{\alpha}$in Eq. 1 is the intensity of the pheromone between cities i to j and α is the parameter that regulate the influence of τ_{ij}. $(n_{ij})^{\beta}$ is the visibility of city j from city I and the β is the parameter to regulate the influence of n_{ij}. In the beginning, k ants are placed to the n cities randomly and then each ant decides the next city to be visited according to the probability P_{ij} given in Eq. 1.

In Eq. 2, t is the time counter, ρ ε[0,1] is to regulate the reduction of τ_{ij}. Δτ is the total increase of trail level on edge (i,j) caused by an ant.

V. EXPERIMENTAL RESULTS

To indicate the capability of the proposed algorithm, it is tested on EIL51 and ST70 from TSPLIB, and the results are compared with optimization results of BSO-DPSO (Hua, Chen, & Xie, 2016) as in Fig. 3, Fig. 4, Fig. 5 and Fig. 6, respectively. In extension, the numerical results of GA-ACO with BSO-DPSO are shown in Table I for EIL51 and for ST70 the comparison is shown in Table II. The proposed algorithm achieves better results and a higher coverage rate than the BSO-DPSO. From Table I and II, it can be seen that the proposed algorithm can find the best fitness value to a best optimal solution for EIL51 that is 478.7 and ST70 that is 937.5 and beneficial to obtain the feasible solutions which are quite close to the best known optimal solution for EIL51 and ST70. Furthermore, the GA-ACO is more stable than the other algorithms.

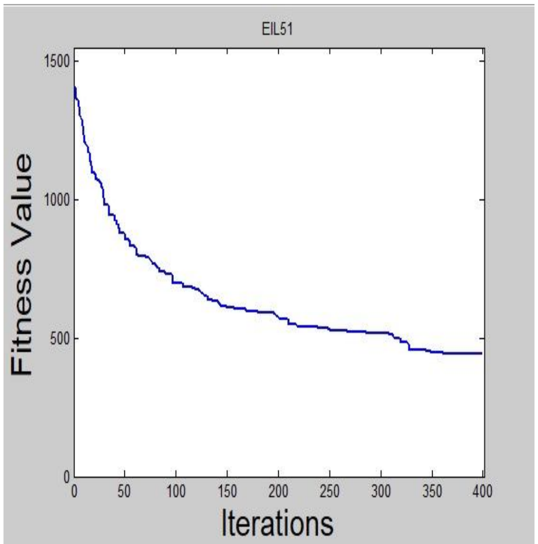


Figure 3 Result of GA-ACO of EIL51 data

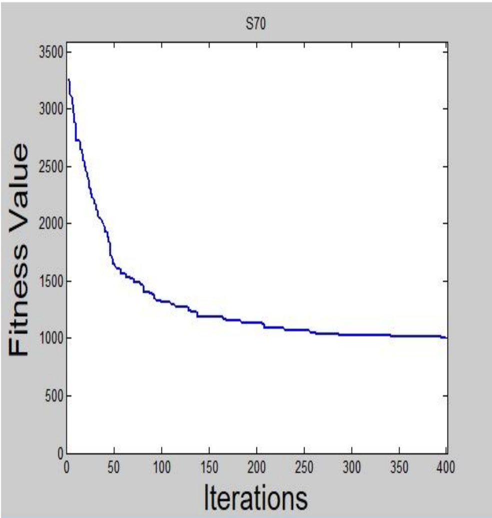


Figure 4 Result of GA-ACO of ST70 data

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VIII, August 2017- Available at www.ijraset.com

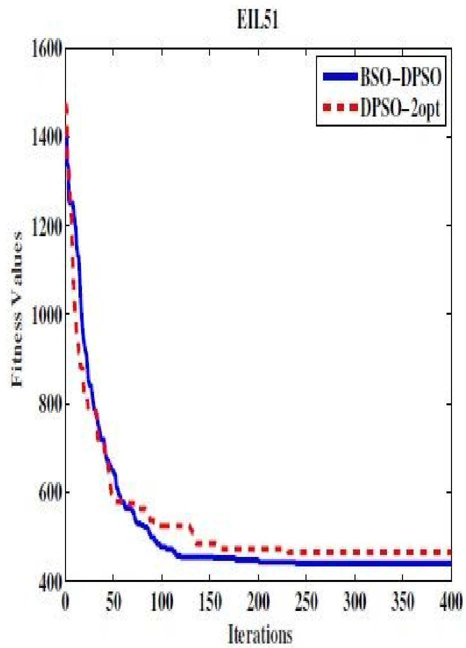


Figure 5 Result of EIL51 of existing algorithms

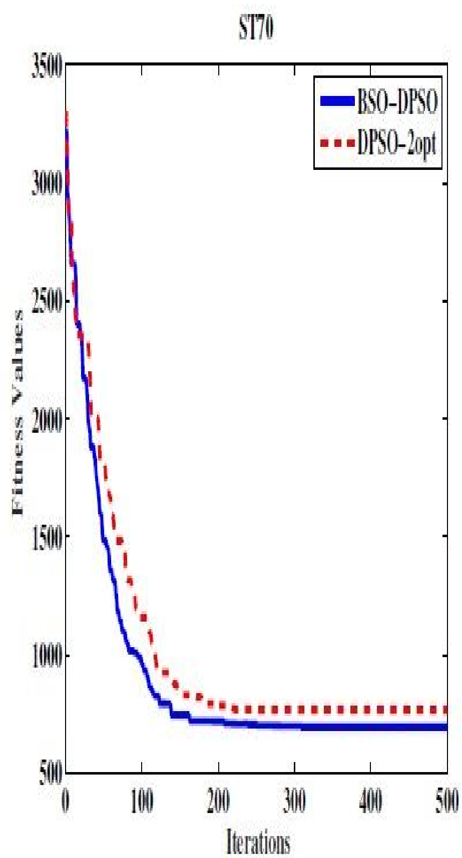


Figure 6 Result of EIL51 of existing algorithms

©IJRASET (UGC Approved Journal): All Rights are Reserved

Experiments with different iterations were conducted to compare the performance of proposed GA-ACO approach with BSO-DPSO. Total fitness value required for EIL51 and ST70 datasets was calculated over entire TSPLIB database. The fitness value at different iterations for different datasets that are considered for comparison is shown in Table I and Table II.

TABLE I
COMPARATIVE ANALYSIS FOR EIL51

S.RO.	Iterations	GA-ACO fitness value	BSO-DPSO fitness value
1	50	890	600
2	100	625	490
3	150	600	455
4	200	525	435
5	250	500	428
6	300	479.5	428
7	350	479	428
8	400	478.7	428

Table I above demonstrates the fitness value required by each algorithm for their best solutions by travelling each and every city exactly once for dataset EIL51.A result of comparison between BSO-DPSO and proposed GA-ACO on the basis of fitness value for the EIL51 dataset is shown in Figure 7.

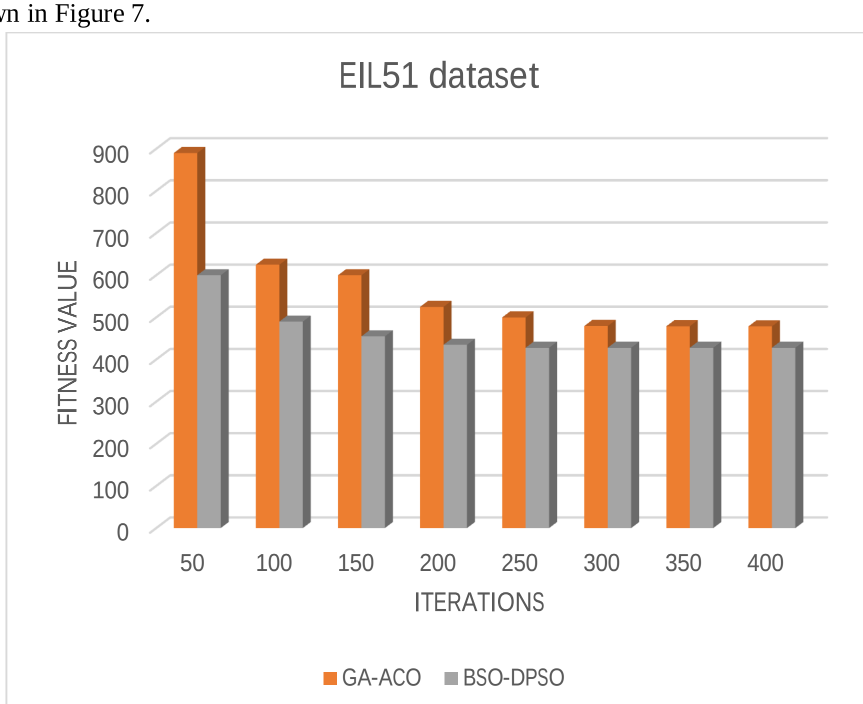


Figure 7 Comparison of Results of GA-ACO, BSO-DPSO for EIL51

It can be concluded from the simulation results that GA-ACO performs better than other considered algorithm in terms of fitness value. GA-ACO has improved fitness value of 478.7 as compared to 428 of BSO-DPSO for an EIL51 dataset for the optimal solution.

©IJRASET (UGC Approved Journal): All Rights are Reserved

Table II below depicts the fitness value enforced by each algorithm by travelling each and every city exactly once for dataset ST70 for their best solutions.

TABLE II
COMPARATIVE ANALYSIS FOR ST70

S.RO.	Iterations	GA-ACO fitness values	BSO-DPSO fitness values
1	50	1800	1625
2	100	1490	1000
3	150	1325	795
4	200	1200	725
5	250	1125	700
6	300	1085	700
7	350	1000	700
8	400	937.5	700

A result of comparison between BSO-DPSO and proposed GA-ACO on the basis of fitness value for dataset ST70 is shown in Figure 8.

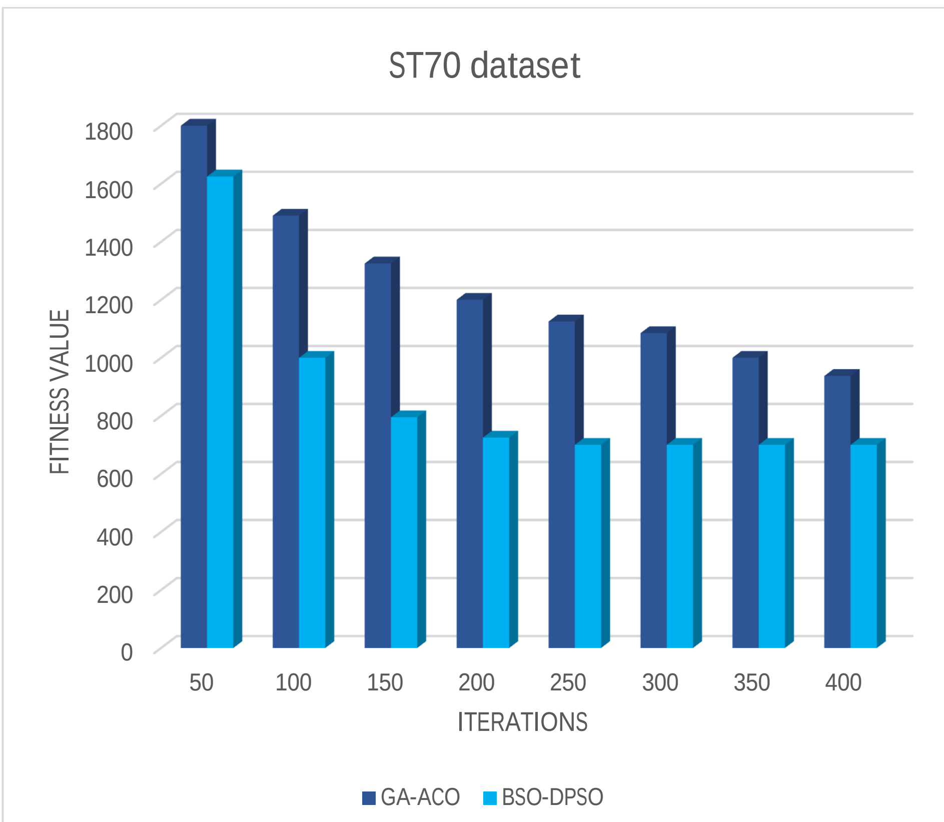


Figure 8 Comparison of Results of GA-ACO, BSO-DPSO for ST70

©IJRASET (UGC Approved Journal): All Rights are Reserved

As seen from the above results GA-ACO computes the fitness value faster than BSO-DPSO. The refined fitness value of 937.15 by GA-ACO has refined fitness value of 937.15 is then compared to 700 fitness value of BSO-DPSO for the EIL51 dataset to get an optimal solution.

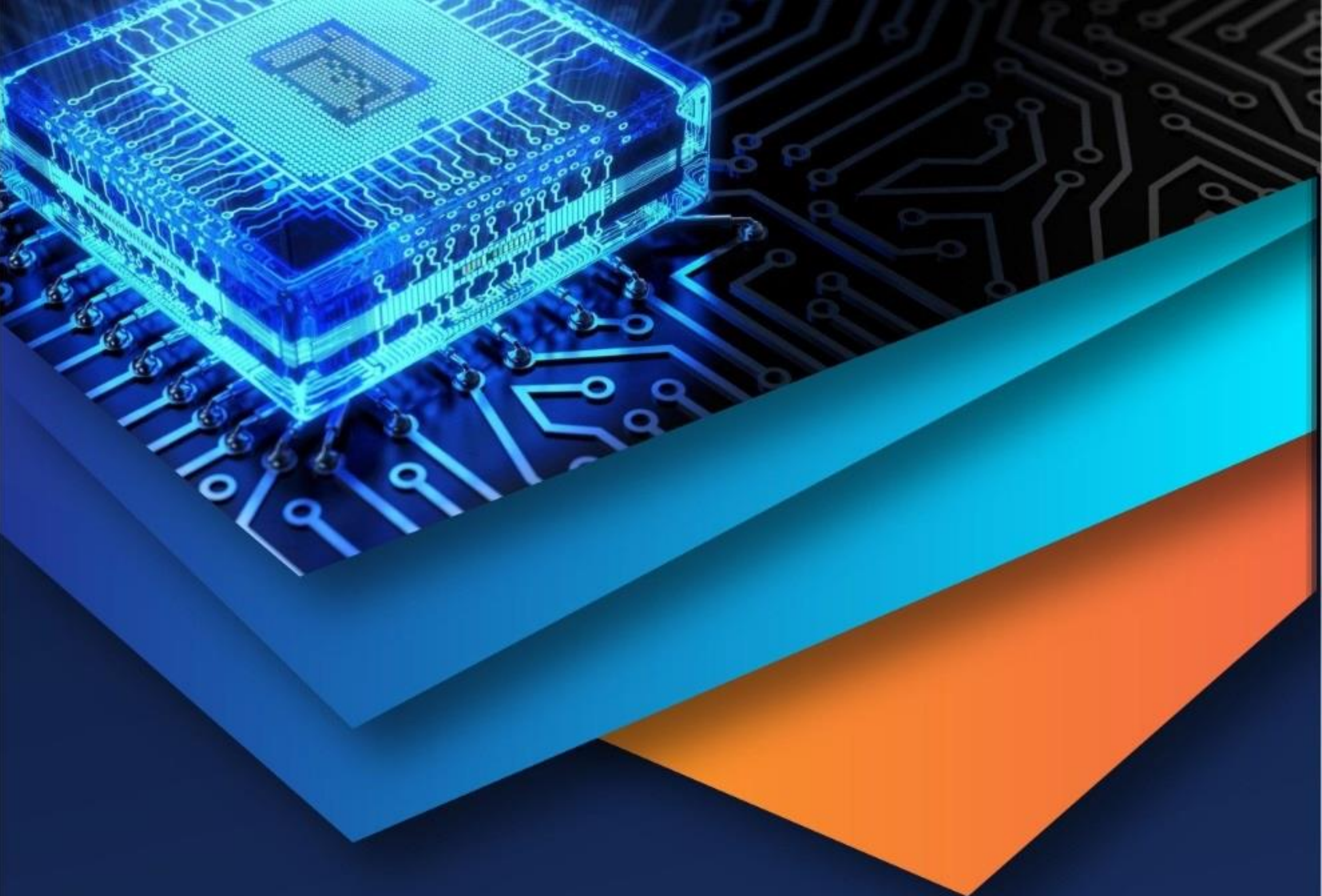
VI. CONCLUSION

In this paper, a new hybrid algorithm, GA-ACO is developed and illustrated in TSP. In GA-ACO, a new probabilistic selection and a crossover are used along with swap mutation and update condition to better hold the needs of discrete problems. In recent years, TSP is a most studied classical problem in computational mathematics and viewed as highly NP-complete combinatorial optimization problems. Here, advancement of GA-ACO is in general form and it can be applied in other discrete problems such as network optimization, graph theory, solid transportation problems, vehicle routing, VLSI chip design, etc. Lastly, it is applied to EIL51 and ST70 from TSPLIB in comparison with the BSO-DPSO. The proposed algorithm attains satisfactory results and assurance a high coverage area in experimental results.

REFRENCES

[1] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," BioSystems, vol. 43, no. 2, pp. 73-81, 1997.
[2] J. Yang, X. Shi, M. Marchese and Y. Liang, "An ant colony optimization method for generalized TSP problem," Progress in Natural Science, vol. 18, no. 11, pp. 1417-1422, November 2008.
[3] G. Dong, W. W. Guo and K. Tickle, "Solving the traveling salesman problem using cooperative genetic ant systems, Expert Systems with Applications, vol. 39, no. 5, pp. 5006-5011, April 2012
[4] V. Dwivedi, T. Chauhan, S. Saxena and P. Agrawal, "Travelling Salesman Problem using Genetic Algorithm," in National Conference on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI), 2012.
[5] D. A. Roy, S. Maity and P. M. Maiti, "Constrained Solid Travelling Salesman Problem," in Informatics, Electronics and Vision (ICIEV), 2015.
[6] S. Kang, S.-S. Kim, J.-H. Won and Y.-M. Kang, "Bidirectional Constructive Crossover for Evolutionary," in 5th International Conference on IT Convergence and Security (ICITCS), 2015.
[7] T. A. S. Masutti and L. N. d. Castro, "TSPoptBees: A Bee-Inspired Algorithm to Solve the Travelling Saleman Problem," in Advanced Applied Informatics (IIAI-AAI), 2016 5th IIAI International Congress, 2016.
[8] Z. Jiang, "Discrete Bat Algorithm for Traveling Salesman Problem," in Information Science and Control Engineering, 2016.
[9] H. Zhou and M. Song, "An Improvement of Partheno-Genetic Algorithm to Solve Multiple Travelling Salesmen Problem," in Computer and Information Science (ICIS), 2016.
[10] Z. Hua, J. Chen and Y. Xie, "Brain Storm Optimization with Discrete Particle Swarm Optimization for TSP," in 12th International Conference on Computational Intelligence and Security, 2016.
[11]

©IJRASET (UGC Approved Journal): All Rights are Reserved



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)