TEST Engineering & Management

May-June 2020
ISSN: 0193-4120 Page No. 546 - 553

A Fast Bee Colony Algorithm to Solve Np-Hard Problems

Nitesh M Sureja
Director, OM Engineering College, Junagadh, Gujarat State, INDIA

Article Info
Volume 83
Page Number: 546 - 553
Publication Issue:
May - June 2020

Article History
Article Received: *11 August 2019*
Revised: *18 November 2019*
Accepted: *23 January 2020*
Publication: *09 May 2020*

Abstract:
The Bee Colony Optimization (BCO), is one the nature inspired algorithms used to solve very complex NP-hard problems very effectively and efficiently. This paper presents an algorithm based on bee colony optimization to solve a combinatorial optimization problem called Random Traveling Salesman Problem (RTSP). We present an algorithm which finds a shortest tour for a traveling salesman with a reduced amount of time required to complete the tour. The results obtained for the randomly generated instances of TSP are then compared with the results obtained with the other very well known combinatorial optimization algorithms. The proposed algorithm produces very good quality solutions in acceptable time in compare to other algorithms.

Keywords: *Meta heuristics, bee colony optimization, combinatorial optimization, swarm intelligence, random traveling salesman problem.*

I. Introduction

In a traveling salesman problem, the salesman has to visit number of cities present in the tour. The salesman has to travel each city exactly once starting from one defined city and reach to this same city for ending his tour. The task of the problem is to find a tour with shortest length. At the same time we must minimize the time required to complete the tour. [1][2]. TSP can be represented as a weighted complete graph in which all nodes represent a city and the arcs represents the path between the cities. We have to find a hamilton cycle with the least weight for a given complete weighted graph. The traveling salesman problem can be illustrated as follows:

TSP = {(A, f, t): $A = (V, E)$ a complete graph,

F is a function, $V \times V \rightarrow Z$,

$b \in Z$,

A is a graph that contains a traveling salesman tour with cost that does not exceed b}.

To understand clearly, show the example shown in figure 1 where a set of 5 cities is shown. The job is to find a shortest path while visiting all the nodes once only and reaching to the starting one. We find two paths; path 1 and path 2. Path 1 is represented as {A, B, C, D, E, A} and the Path 2 is represented as {A, B, C, E, D, A}. While calculating of both the paths, path 1 is best with the distance 22.

TSP is widespread in engineering applications and some industrial problems such as machine scheduling, cellular manufacturing and frequency assignment problems can be formulated as a TSP. The TSP is very well known NP-hard problem. No exact algorithm can solve it in acceptable time. The time required to solve traveling salesman problem is exponential. So, normally we use combinatorial optimization algorithms to find a near to optimal solution of such problems. This type of algorithms is also known as metaheuristic algorithms. Many metaheuristic algorithms applied to solve traveling salesman problem successfully. There are many ways to classify the algorithms based on the ability to produce optimal solutions. They are classified as exact and heuristic algorithms. Exact algorithms produce result in exponential time but at the same time it fails to solve nearly all combinatorial problems in exponential time. While heuristic algorithms give near to optimal solution of all kind of optimization problems in acceptable time.

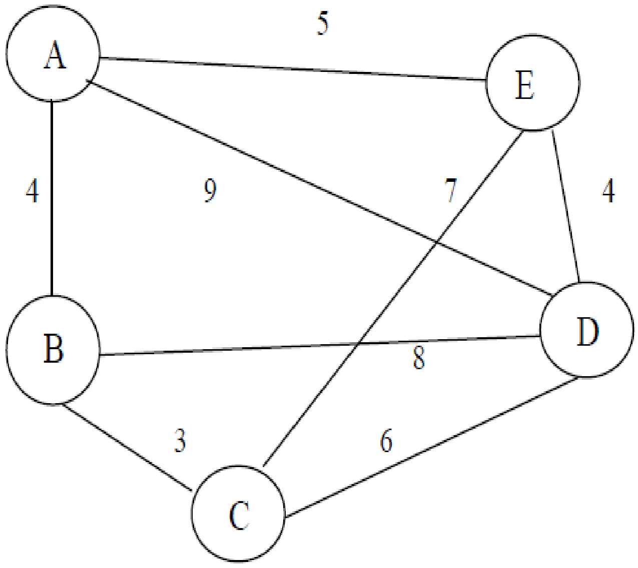

Figure 1: A Graph with weights on its edges

The brute force method [3], Greedy Algorithm [4], branch and bound algorithm [5], cutting plane algorithm [6] and dynamic programming [7] are some of the best known exact algorithms. These methods produce the good results for less complex optimization problems like a TSP with very less number of cities for traveling. So it is necessary to use heuristic algorithms where the complexity of the problem is very high. Usually heuristic algorithms produce the optimal solutions in the given acceptable time. But, this is not the case every time means there is no guarantee of good quality solution to the given problem every time. Heuristic algorithms focus on solving the special type of problems.

Heuristics algorithms comprises of the stochastic and deterministic composition. They usually determined to find a global optimal solution removing the problems of local optima. The heuristic algorithms are designed in such a way that they can be used to solve any kind of optimization problems. They normally show the way to apply required procedures to turn into solution of the problem. The group of the most known metaheuristics includes: evolutionary algorithms[8], genetic algorithms[9], particle swarm optimization[10], differential evolution[11], ant colony optimization[12], harmony search[13], artificial immune algorithms[14], bat algorithms[15], firefly algorithms[16], cuckoo search[17], artificial buffalo optimization[18] etc.

In this research we solve the Random Travelling Salesman Problem (RTSP) using the basic bee colony optimization (BCO) algorithm. BCO is used to generate the optimized path for randomly generated TSP instances. To check the efficiency of our proposed solution we compared our solution with the solution generated with other algorithms like, Memetic Algorithm, Firefly Algorithm and Particle Swarm algorithm.

II. RANDON TRAVELING SALESMAN PROBLEM

In a Random Traveling Salesman Problem, city problems are generated randomly. This is done to explore more search space to address the problem of local optima. A random problem instance generator is used to generate the city problems randomly [19][20].

Other than random TSP, there are many types of traveling salesman problem (TSP) described in the literature. To list a few; Symmetric TSP, Asymmetric TSP, Dynamic TSP, Spherical TSP etc. Symmetric TSP is a tsp where distance between the cities is same from the either side. Asymmetric TSP is a TSP where distance between the cities from the either side is not same. Dynamic TSP is a TSP where the problem changes itself at run time. In a spherical TSP all cities lie on a sphere. This paper presents a bee colony optimization algorithm to solve Random Traveling Salesman Problem. All city problems are generated in the range of 10 to 100.

III. BEE COLONY OPTIMIZATION

Introduction

The Bee Colony Optimization (BCO) algorithm is a Nature Inspired Algorithm which is inspired by foraging behavior of honeybees. The Bee Colony algorithm is proposed by Lucic and Teodorovic

May-June 2020
ISSN: 0193-4120 Page No. 546 - 553

[21][22][23]. The BCO builds a multi agent system which in turn solves optimization problems very successfully. The artificial bee colony follows basic principles followed by bee colonies naturally. This metaheuristic has been successfully applied to various large and complex real world problems. It has been found that Bee Colony Algorithm posses the capability of finding the quality solutions of combinatorial problems within acceptable computer time [24].

The BCO uses a random search technique stochastically. This technique uses similarities of foraging behaviour of bees in nature and the way in which metaheuristic algorithms look for a best possible solution for combinatorial problems. The BCO builds a artificial bee colony capable to solve tough optimization problems. Artificially created bees explore the search area in search of feasible solutions. Self-directed artificial bees work together and switch over information to find better solutions. Artificially created bees focuses on more potential areas and start to throw away solutions from the less potential areas sharing the knowledge achieved by working collectively. The Bee colony algorithm runs iteratively until the stopping criteria are satisfied. The BCO works in a decentralized way with their self organized nature which provides a basis for parallelization. It also has a capability for keeping away itself from local minima [24].

Bee Colony Algorithm

The Bee Colony Algorithm is a population based optimization algorithm. In whichn a population of artificial bees works collectively to find an optimal solution. Each bee in a population represents a solution which in turn generates other solution for the problem. The bee colony algorithm used to passes known as forward pass and backward pass which works alternatively to generate a single step of the algorithm. Each artificial bee explores the search area using defined moves for creating and/or improving the solution, which gives a new solution I its forward pass [24].

In the backward pass, bees go to back to hive and start their second phase using the information of partially obtained solutions. In this pass sharing the information about the solutions achieved by all artificial bees is done. At the end of one step all the solution achieved by bees are evaluated based on objective function. One probability value associated with each bee decides loyalty of bees. The bees which produce better solutions have more chances to keep their solution in the population for next steps. The bees which got success to keep their solutions are known as recruiter bees. All the other bees which abandon their solutions are known as uncommitted bees. All the Bees which generate better solutions take part in the next step using a probability value. All bees are divided into two groups known as recruiters and uncommitted bees [24].

The algorithm executes itself in two phases, forward and backward pass which are executed alternatively for finding optimal solution. After the evaluation of all solution, a best one is found which is used to update the global best solution and an iteration of the algorithm is completed. In this way bee colony algorithm runs in iterations until stopping criteria is satisfied. We can use number of iterations, CPU time, stagnation etc as the stopping criteria. The beat found solution at the time of reaching at stopping criteria will be the final best optimal solution [24].

IV. THE PROPOSED BCO-RTSP MODEL

We can't apply basic BCO directly to the TSP as it is a discrete optimization problem. Tedrovic [23] has given a frame work for the TSP by modifying the basic BCO for TSP. we will use this frame work here to solve our Random Traveling Salesman problem.

Bee colony optimization (BCO) [21][22][23][25][26] creates a colony of artificial bees which in turn used to solve NP-hard problems. We apply BCO to traveling salesman problem to find the shortest tour iteratively. In Bee Colony Optimization algorithm, a bee starts to find the food without any knowledge of food source. This bee is known as unemployed bee. The Bee explores one

May-June 2020
ISSN: 0193-4120 Page No. 546 - 553

food source (nectar) to other food source (nectar), completes one stage and returns to its home (hive). This process is known as forward pass. A completed tour of a bee comprises of stages made of a collection of food sources. A food source (nectar) is represented as a city in the model designed for TSP and a stage of the solution works as an element of the tour. A probability value is used by Teodrovic which is used by a bee k selects the next city j from city i in the stage next of stage u in iteration z is given by equation-1[13][27][28].

$$P_{ij}^{k}\,(u+1,z) = \begin{cases} \dfrac{e^{-ad_{ij}\frac{z}{\sum_{h}^{z-1} n_{ij(h)}}}}{\sum_{l\in N_{k(u,z)}} e^{-ad_{il}\frac{z}{\sum_{h}^{z-1} n_{il(h)}}}} & i = g_{k}{}^{(u,z),\, j\in N_{k(u,z)}},\ \forall k,u,z \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

Here:

1. h represents maximum of (z-b,1),
2. Indexes of nodes is represented by i,j otherwise
3. Distance between a link (i,j) is presented by dij
4. Index of bee is represented by k.
5. B represents number of bees in the hive,
6. Index of iterations is represented by z,
7. M represents maximum allowed iterations,
8. s represents nodes visited by bees in a stage,
9. Stage index is represented by u,
10. nil(h) gives the number of bees visit link (i,1) in iteration number h. b represents the length of the memory.
11. Last node visited by k bee in u stage during iteration z is represented by gk(u,z).
12. Set of all unvisited nodes represented by a set for k bee in u stage during iteration z is designated as Nk(u,z),
13. a represents a parameter specified by analyst

using equation-1, we can find that the city j which has the greater distance from city i has the worst chance to move to city j from i. here, if algorithm uses the more iterations z shows that the distance has a heavy influence. More number of iteration also shows that the bee has a limitation in exploring the solutions. As an idea, some bees starts their foraging in the first phase, some starts their foraging in the second phase, some starts in third phase, etc. a bee remains active till the completion of a iteration. Each bee uses a probability to select the city for traveling.

All bees come back to home and unload and store the food (nectar) after completing one stage. This process is known as backward pass. In this pass, we have three types of bees; Scout, follower and recruiter bees. Scout bees keep the previous step and go ahead for the further steps without interacting to the others. Follower bee abandons the nectar and follows the others based on the waggle dance of recruiter bees. Recruiter bees keep their previous step and recruit follower bees for further process.

To categorize these three of type bees, Bee Colony Algorithm uses Eq.2 [27][28] for categorizing these three types of bees and to determine the probability which a bee *k* in next stage (*u+1*) uses the same partial tour in stage *u* in iteration *z* as described below:

$$P_{K}(u+1,z) = e^{-\frac{L_{k}(u,z) - min_{r\in W(u,z)}(L_{r}\,(u,z))}{uz}} \tag{2}$$

We use inverse of the tour length of each path as the fitness function value to find the best solution. Flowchart of proposed BCO-RTSP model is given below.

May-June 2020
ISSN: 0193-4120 Page No. 546 - 553

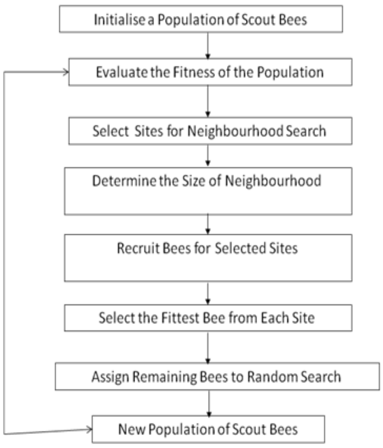

Figure 2: Flowchart of proposed BCO-RTSP Model

V. IMPLEMENTATION AND RESULTS

Proposed BCO-RTSP model is implemented in Matlab -15. and executed on a Intel core i3 seventh generation machine with eight GB RAM. As mentioned, all TSP problems are generated randomly in the range of 10 to 100. Algorithm executed itself based on a termination criterion. Termination criteria used here is number of iterations. Table 1 shows the results obtained by the algorithm. Some of the Results are also shown graphically in the figure 3 to 11

Table 1: Results Obtained from BCO-RTSP Model with other algorithms

City Problem	MA		FA		PSO		BCO	
	Length	*Time*	*Length*	*Time*	*Length*	*Time*	*Length*	*Time*
10	300.75	7.13	300.75	13.45	281.6978	169.741	266.2837	22.3350
20	372.34	8.46	372.34	14.93	406.4415	190.053	346.8796	89.2720
30	408.57	13.60	409.01	17.34	452.2936	220.370	510.6232	228.7319
40	537.95	14.78	523.96	23.18	505.4753	258.914	518.3001	398.9980
50	575.36	11.64	545.54	19.82	609.8611	309.500	595.3425	622.4317
60	616.31	25.74	622.26	25.45	720.5503	389.281	660.2627	886.5954
70	723.90	14.36	661.64	22.47	785.1615	439.166	681.6139	917.1323
80	768.69	17.43	716.51	26.29	902.1146	515.915	772.5417	943.5725
90	757.38	20.87	731.68	31.89	943.2347	602.499	811.4022	964.7341
100	875.87	20.59	806.24	43.95	980.1618	685.272	790.8024	989.5345

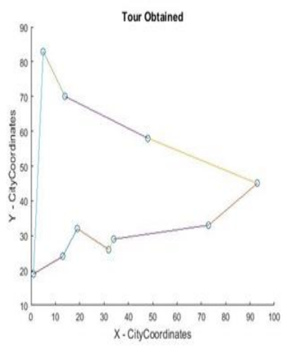

Figure 3: Results for 10 City Problem

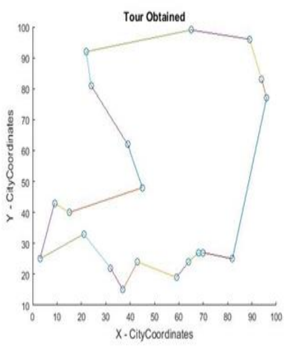

Figure 4: Results for 20 City Problem

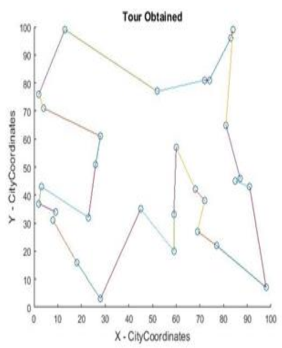

Figure 5: Results for 30 City Problem

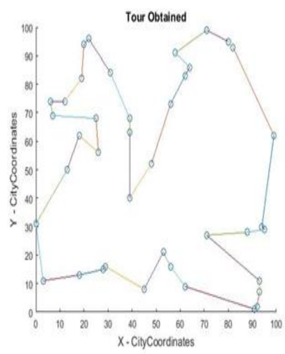

Figure 6: Results for 40 City Problem

May-June 2020
ISSN: 0193-4120 Page No. 546 - 553

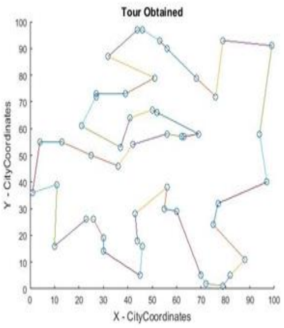

Figure 7: Results for 50 City Problem

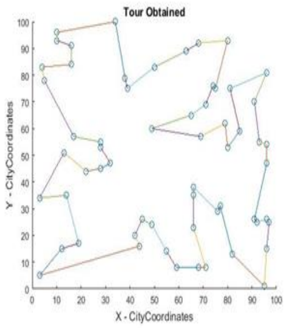

Figure 8: Results for 60 City Problem

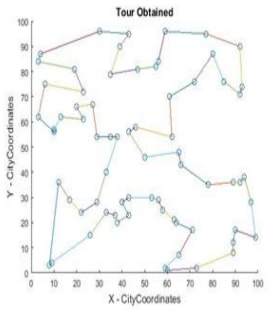

Figure 9: Results for 70 City Problem

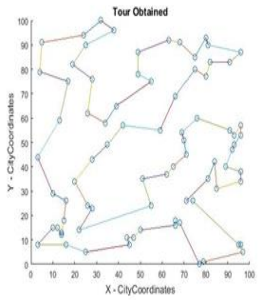

Figure 10: Results for 80 City Problem

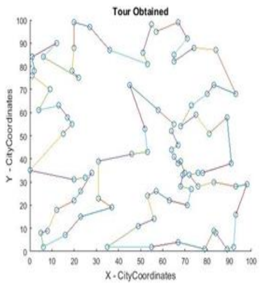

Figure 11: Results for 90 City Problem

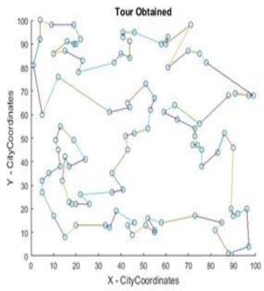

Figure 12: Results for 100 City Problem

VI. CONCLUSIONS

A Bee Colony algorithm based on the food foraging behaviour of bees is presented to solve a NP –Hard problem known as Random TSP. The model has been tested on a set of randomly generated TSP problems. The algorithm generates very good results in terms of quality and distance taking little bit more time to converge in a solution. It is observed that the convergence time increases with the increase of complexity of the problem. It is also observed that with the some parameter settings the time can be improved for the big size problems also. It is concluded that Bee colony algorithm can solve a wide variety of real life applications.

VII. ACKNOWLEDGEMENTS

Authors are greatful to OM engineering College – Junagadh for providing all kinds of assistance to do this research work.

VIII. REFERENCES

1. G. Dantzig, R. Fulkerson, S. Johnson, Solution of a Large-Scale Traveling Salesman Problem, J. Oper. Res. Soc. 2 (1954) 393–410.
2. Gerhard Reinelt. "The Traveling Salesman: Computational Solutions for TSP Applications.", Springer-Verlag, (1994),Berlin, Heidelberg
3. http://www.cs.sfu.ca/CourseCentral/125/tjd/tsp_example.html
4. Eugene Lawler, L., Lenstra, J.K., Rinnooy Kan, A.H.G, and Shmoys, D.B., 1985. The Traveling Salesman Problem, John Wiley & Sons.
5. Little J. D. C., Murty K. G., Sweeney D. W and Karel C., 1963. An algorithm for the traveling salesman problem . Opns Res 11: 972–989
6. G. Dantzig, R. Fulkerson, S. Johnson, Solution of a Large-Scale Traveling Salesman Problem, J. Oper. Res. Soc. 2 (1954) 393–410.
7. Martin G. T., 1966. Solving the traveling salesman problem by integer programming. Working Paper, CEIR, New York.
8. Tsai, H.-K., Yang, J.-M., Tsai, Y.-F., & Kao, C.-Y. (2004). An Evolutionary Algorithm for Large Traveling Salesman Problems. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 34(4), 1718–1729
9. Abid Hussain, Yousaf Shad Muhammad, M. Nauman Sajid, Ijaz Hussain, Alaa Mohamd Shoukry, and Showkat Gani, "Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator," Computational Intelligence and Neuroscience, vol. 2017, Article ID 7430125,

7 pages, 2017.
10. G. Goldbarg, E. F., C., M., & de Souz, G. R. (2008). Particle Swarm Optimization Algorithm for the Traveling Salesman Problem.
11. Mi, M., Huifeng, X., Ming, Z., & Yu, G. (2010). An Improved Differential Evolution Algorithm for TSP Problem. 2010 International Conference on Intelligent Computation Technology and Automation.
12. M. Dorigo, L. Gambardella, “Ant colonies for the Traveling salesman problem.” Biosystems 43 (1997): (73-81).
13. X.S. Yang, “Harmony Search as a Metaheuristic Algorithm”, Studies in Computational Intelligence, Springer Berlin, Vol. 191, pp. 1-14 (2009)
14. M. Bakhouya, Jaafar Gaber,(2007), “An Immune Inspired-based Optimization Algorithm: Application to the Traveling Salesman Problem”, Advanced Modeling and Optimization, Volume 9, Number 1, 105-116.
15. Saji, Y., Riffi, M.E. A novel discrete bat algorithm for solving the travelling salesman problem. Neural Computing & Applications, 27, (2016) :(1853–1866)
16. Jati, G. K., & Suyanto. (2011). Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem. Lecture Notes in Computer Science, 393–403.
17. Jati GK, Manurung HM, Suyanto. “Discrete cuckoo search for traveling salesman problem”, In Proceedings – 7th International Conference on Computing and Convergence Technology, 2012. p. 993-997.
18. Odili, J. B., & Mohmad Kahar, M. N. (2016). Solving the Traveling Salesman’s Problem Using the African Buffalo Optimization. Computational Intelligence and Neuroscience, 2016, 1–12.
19. Nitesh M Sureja, Bharat V Chawda, Random Travelling Salesman Problem using Genetic Algorithms”, IFRSA’s nternational Journal Of Computing, Volume 2, Issue 2, April 2012.
20. Nitesh M. Sureja, Bharat V. Chawda, “Memetic Algorithm a Metaheuristic Approach to Solve RTSP”, IJCSEITR, ISSN 2249- 6831, Vol. 3, Issue 2, pp. 183-186, June 2013
21. P. Lucic, D. Teodorovic, ”Transportation Modeling: An Artificial Life Approach”,Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence, 2002.
22. P. Lucic, ”Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing, PhD Thesis, Civil Engineering, Faculty of the Virginia Polytechnic Institute and State University, 2002.
23. D. Teodorovic, P.Lucic, G. Markovic, M. DellOrco, ”Bee Colony Optimization: Principles and Applications”, 8th Seminar on Neural Network Applications in Electrical Engineering, NEUREL, 2006
24. http://old.sf.bg.ac.rs/index.php/sr-RS/katedre/59-katedra-za-operaciona-istrazivanja-u-saobracaju/814-bee-colony-optimization-bco
25. L.P. Wong, M.Y.H. Low, C. S. Chong, ”A bee colony optimization algorithm for travelling salesman problem”, in Proceedings of Second Asia International Conference on Modelling and Simulation (AMS), pp. 818-823, 2008
26. Girsang, A. S., Tsai, Chaudhuri, Buddhadeb. (2007).Forest And Tribals A historical Review of Forest Policy, Forest .In Dr. Chittarajan Kumar Pathy (Ed.), Forest, Government and the Tribe, (pp.1-17). New Delhi: Concept of Publishing Company C.-W., & Yang, C.-S. (2012). A Fast Bee Colony Optimization for Traveling Salesman Problem. 2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications.
27. Kocer, H. E., & Akca, M. R. (2014). An improved artificial bee colony algorithm with local search for traveling salesman problem. Cybernetics and systems, 45(8), 635–649.
28. Majida Ali Abed, " A Metaheuristic Algorithm ACO of Handwritten Arabic Characters Recognition ", International Journal of Computer Science Engineering

May-June 2020
ISSN: 0193-4120 Page No. 546 - 553

and Information Technology Research (IJCSEITR), Vol. 3, Issue 3, pp. 151-168

29. Priyanka Lonkar & D.C. Mehetre, " Data Forwarding Based on Interest Using ABC Algorithm in SAN ", International Journal of Computer Networking, Wireless and Mobile Communications (IJCNWMC), Vol. 5, Issue 4, pp. 9-18
30. G. Umashankar & D. Saravanan, " Single Objective for an Integer Partial Flexible Open Shop Scheduling Problem Using Developed Ant Colony Optimization ", IJMPERD, Vol. 8, Issue 3, pp. 1121-1132
31. Moein Khosravi, Milad Askari Hashemabadi, Vajihe Sharifi Davarani, Mohadeseh Sharifi Davarani & Mahdi MozaffariLegha, " Loss Reduction with Optimization of Capacitor Placement Using BFA Algorithm - Case Study for a 20 KV Network in Iran ", International Journal of Electrical and Electronics Engineering (IJEEE), Vol. 3, Issue 3, pp. 69-80
32. M. Padmavathi & Shaik Mahaboob Basha, " A Conceptual Analysis on Cloud Computing ACO Load Balancing Techniques ", International Journal of Computer Science and Engineering (IJCSE), Vol. 6, Issue 4, pp. 39-48
33. Jalal A. Sultan & Raghad M. Jasim, " Demand Forecasting using Artificial Neural Networks Optimized by Artificial Bee Colony ", BEST: International Journal of Management, Information Technology and Engineering (BEST: IJMITE), Vol. 4, Issue 7, pp. 77-88

AUTHORS PROFILE

Dr. Nitesh M. Sureja is a Diploma, B.E., M.E and PHD, all in Computer Engineering. He has an industrial experience of four years in Raj Infotech as a Sales and Service engineer of Computer Systems and Peripherals. He worked in CCET Wadhwan City (Saurashtra University) as Lecturer. He also worked in SPCE, Visnagar (Hemchandracharya North Gujarat University) as Lecturer and incharge of the Department of Computer and IT. He worked as Associate Professor (Computer Engineering) at G.H. Patel College of Engineering and Technology, Vallabh Vidyanagar for eight years. . He worked as a principal at C. U. Shah Technical Institute of diploma studies, Wadhwan for two years. Currently he is working as a director at OM Engineering College, Junagadh, Gujarat. He is a life member of ISTE and member of IEANG and IMETE. He has been offering his services as reviewer of various national and international Journals such as IJCSI and UBICC, and reviewer of various international and national conferences. He has guided more than 50 UG Projects and 2 PG. dissertations. He has published 4 International Journal papers and 3 Conference papers. His fields of interest and research are Swarm Intelligence, Nature and Bio inspired Computing and Artificial Intelligence.