Dynamic Simulated Annealing for solving the Traveling Salesman Problem with Cooling Enhancer and Modified Acceptance Probability

Akshay Vyas, Dashmeet Kaur Chawla, Dr. Urjita Thakar

Department of Computer Engineering, Shri G.S. Institute of Technology & Science, 23 Park Road, Nr. Lantern Square, Indore, Madhya Pradesh, India, PIN 452003

DOI: 10.29322/IJSRP.8.3.2018.p7531
http://dx.doi.org/10.29322/IJSRP.8.3.2018.p7531

Abstract- In this paper, a dynamic (i.e. self-adaptive according to the number of nodes) Simulated Annealing Algorithm is presented to solve the well-known Traveling Salesman Problem (TSP). In the presented algorithm, the temperature parameter is adjusted on the basis of the number of nodes. To achieve dynamicity, a new parameter named "Cooling Enhancer" is introduced to control the cooling rate, thereby, regulating the temperature. Additionally, an enhanced version of acceptance probability has been used. The efficacy of Dynamic Simulated Annealing with Cooling Enhancer & Modified Acceptance Probability (DSA-CE&MAP) is compared against the basic simulated annealing algorithm (SA) [2] for some benchmark TSPLIB instances [1]. Experimental results illustrate that the new dynamic simulated annealing algorithm performs better than the basic simulated annealing algorithm for solving TSP. It has been observed that the quality of solutions (i.e. minimum total cost or distance) is significantly increased as compared to earlier method.

Index Terms- Traveling Salesman Problem, NP-complete, Simulated Annealing Algorithm, Acceptance Probability, Temperature, Cooling rate, Cooling Enhancer.

I. INTRODUCTION

The Traveling Salesman Problem (TSP) is one of the archetypes and archaic problems in Computer Science and Operations Research. It can be stated as:
A network with 'n' cities (or nodes) with 'node 1' as 'source' and a travel expense (or distance, or travel time etc.,) matrix $C= [c_{ij}]$ of order n associated with ordered node pairs (i, j) is given. The problem is to find a least cost Hamiltonian cycle.

On the basis of the structure of the cost (or expense) matrix, the TSPs are classified into two groups – symmetric and asymmetric. The TSP is symmetric if $c_{ij} = c_{ji}$, ∀ i, j and asymmetric otherwise. For an n-city asymmetric TSP, there are $(n-1)!$ possible solutions, one or more of which gives the minimum cost. For an n-city symmetric TSP, there are $\frac{(n-1)!}{2}$ possible solutions along with their reverse cyclic permutations having the same total cost. In either case, the total number of solutions becomes extremely humongous for a moderate number of nodes, making the exhaustive search non-viable.

TSP has captivated the attention of many researchers and remains an active research area. It is a proven NP-Complete problem [3]. A large number of real-world problems can be modelled by TSP. Some of them are:- Drilling of printed circuit boards and threading of scan cells in a testable VLSI circuit [4], X-ray crystallography [5], Overhauling gas turbine engines [6], Computer wiring [6], Vehicle routing [6], Mask plotting in PCB production [6], Warehouse automation system [6].

All practical applications require solving larger problems, hence emphasis has shifted from the aim of finding exactly optimal solutions for TSP to the aim of getting, heuristically, 'better solutions' in a reasonable time and 'establishing the degree of goodness'. Several intelligent algorithms are available to solve the TSP, some of them are:- artificial neural network [7], genetic algorithms [8], simulated annealing algorithm [9], ant colony optimization algorithm[10], particle swarm optimization [11], consultant-guided search algorithm [12] and many more. Simulated Annealing Algorithm (SA) is one of the metaheuristic search algorithms that have been

used widely to solve the TSP instances. The basic version of SA algorithm is not good in terms of quality of solutions (i.e. minimum total cost or distance). Therefore, an improved simulated annealing algorithm has been presented in this paper. It uses a new parameter, "Cooling Enhancer" to control the cooling rate in order to regulate the temperature (i.e. system energy) and also employs a modified acceptance probability. The presented algorithm has been found to produce better quality of solutions.
The paper is structured as follows: Section 2 provides a background about the simulated annealing algorithm. In Section 3, the related work in the sphere is propounded. Then, the proposed approach to solving the TSP is proffered in Section 4. The experimentation and results are given in Section 5. Finally, the paper is concluded in section 6.

II. BACKGROUND STUDY

The SA algorithm is one of the efficient methods for the continuous and discrete optimization problems. It is derived from the simulation of the cooling schedule of metals. The cooling process is controlled by a defined function which is convenient to implement. The word "Annealing" is referred to as tempering certain alloys of metal, glass, or crystal by heating above its melting point, holding its temperature, and then cooling it very slowly until it solidifies into a perfect crystalline structure. This physical/chemical process produces high-quality materials. The simulation of this process is known as simulated annealing (SA) [16]. There is an analogy of SA with an optimization procedure. The defect-free crystal state corresponds to the global minimum energy configuration (for TSP, tour with the minimum cost). The physical material states correspond to problem solutions (for TSP, all possible tours), the energy of a state to cost of a solution (for TSP, the tour cost), and the temperature to a control parameter.
The SA algorithm is not used for initial solutions and has been discerned to have a bad performance and slow convergence when applied to the complex TSP.

III. RELATED WORK

In this section, the earlier work related to the field is discussed. The Basic Simulated Annealing Algorithm [2] has been improved by several researchers for solving the TSP. Liu et al introduced SA integrated with the Tabu search in order to achieve better solutions. The temperature was reduced adaptively with a temperature control function [13]. Based on most of the edges in the best circuit linked by neighbour cities, the probabilistic neighborhood model was introduced by Li [13] and merge into the optimization process of the BSA algorithm. The SA algorithm was also integrated with the ant colony optimization [14] to utilize their advantages together. In order to speed up the convergence of the BSA and obtain the better approximate solutions (i.e. Hamiltonian cycles), the four vertices and three lines inequality was merged into the optimization process of the BSA using the four-point conditions for symmetrical TSP which has been summarized by Vladimir [15]. When the Hamiltonian circuits are generated with the BSA, the inequality are applied to the local Hamiltonian paths in the Hamiltonian cycles.
The algorithm for Basic Simulated Annealing [2] for solving TSP is as follows:-

Step 1: Initialize cooling_rate with a small value such as 0.001.
Generate an initial random tour x.

Step 2: Initialize T with a large value such as 100000.

Step 3: Repeat:

i. Generate next tour $(x + \Delta x)$ by applying some operations on the current tour x.
ii. Evaluate $\Delta E(x) = E(x + \Delta x) - E(x)$, (i.e. neighborTourCost - currentTourCost):
if $\Delta E(x) < 0$, keep the new state (i.e. new path distance less than current distance);
otherwise, accept the new state with acceptance probability $P = e^{-\Delta E/T}$.
iii. If $E_{bestSoFar}(x) > E(x + \Delta x)$, then set $E_{bestSoFar}(x) = E(x + \Delta x)$.

iv. Set $T = T - \Delta T$, $\forall$ $\Delta T = T$ x cooling_rate.
until T is small enough.

One of the issues with the basic SA is that it is not adaptive to the problem size i.e. the temperature change is independent of the number of nodes (or cities). On running the basic SA for different TSPLIB instances, it is observed that if the temperature is decreased fast or slowly for small number of nodes, then, in either case, the results are almost similar; however, if the number of nodes is large (say more than 100), then it gives better results on decreasing the temperature slowly. Moreover, the basic SA will run for almost the same time for problems of all size and hence produce the good results for a small size of nodes and bad results for problems with a moderate or large number of nodes. Moreover, it would be the duty of the operator to decide concerning how much time the algorithm should be run to get best result in a feasible amount of time, which could be cumbersome.

IV. PROPOSED APPROACH

The proposed approach, Dynamic Simulated Annealing with Cooling Enhancer & Modified Acceptance Probability (DSA-CE&MAP) is developed by enhancing the parameters of the basic simulated annealing algorithm, temperature, cooling rate and acceptance probability to produce better solutions.

For the algorithm to adapt and adjust itself to the number of nodes, a parameter named as "Cooling Enhancer" is introduced which controls the "cooling_rate", thereby controlling the decrease in temperature according to the number of nodes (or cities) after each iteration.

Additionally, for accepting the less good solutions (i.e. solutions with somewhat high total cost or distance), the acceptance probability is modified to engender Modified Acceptance Probability (MAP) which significantly contributes to the better results and prevents the solutions in getting caught in local minima. Modified acceptance probability helps in reducing the acceptance probabilities of the tours that have cost much larger than the current best tour cost.

The algorithm for Dynamic Simulated Annealing with Cooling Enhancer and Modified Acceptance Probability (DSA-CE&MAP) for solving TSP is as follows:-

Step 1: Initialize cooling_rate with a small value such as 0.001.
Generate an initial random tour x.

Step 2: Initialize T with a large value such as 100000.

Step 3: if totalCities < 30, then set coolingEnhancer = 0.5.
else if totalCities < 150 then set coolingEnhancer = 0.05.
else if totalCities < 750 then set coolingEnhancer = 0.005.
Otherwise, set coolingEnhancer = 0.0005.

Step 3: Repeat:

i. Generate next tour $(x + \Delta x)$ by applying some operations on the current tour x.
ii. Evaluate $\Delta E(x) = E(x + \Delta x) - E(x)$, (i.e. neighborTourCost - currentTourCost):
if $\Delta E(x) < 0$, keep the new state (i.e. new path distance less than current distance);
otherwise, evalutate $\Delta E` = E_{bestSoFar}(x) - E(x + \Delta x)$, (i.e. bestTourCost - neighborTourCost) and then accept the new state with
acceptance probability, $P = \frac{e{-}\Delta E/T}{e{-}\Delta E`/T}$
iii. If $E_{bestSoFar}(x) > E(x + \Delta x)$, then set $E_{bestSoFar}(x) = E(x + \Delta x)$.
iv. Set $T = T - \Delta T$, ∀ ΔT= T x coolingEnhancer x cooling_rate.
until T is small enough.

The effects of Modified Acceptance Probability (MAP) parameter for DSA-CE&MAP algorithm can be illustrated on br17(an Asymmetric TSPLIB instance) as follows :

For each iteration, the values for three parameters are evaluated:- (i) currentTourCost (the total tour cost of the current path), (ii) bestTourCost (the total tour cost of the smallest path found till now), and (iii) neighborTourCost (the total tour cost of the next considered path). The acceptance probability is calculated only if, *ΔE(x)= (neighborTourCost − currentTourCost)* > 0. For a random iteration, let the current path be (1,4,8,17,14,10,6,11,2,12,9,3,5,16,13,7,15) with tour cost 241 and the best tour path be (1,9,15,8,17,12,7,4,5,16,10,14,11,6,13,3,2) with tour cost 85 and temperature be 1000. Now, let the neighbor path under consideration be (1,9,8,2,17,7,14,6,11,16,12,10,5,3,4,13,15) with total cost 406 and another neighbor path be (1,4,8,17,13,16,5,3,9,12,2,11,6,10,14,7,15) with tour cost 260. Then, the acceptance probability of these paths with basic SA and DSA-CE&MAP is obtained as given in the following table:

TABLE I: Acceptance probability as calculated by the Basic SA and DSA-CE&MAP for a given instance

br17 (a TSPLIB instance)	Basic SA	DSA-CE&MAP
Case (i) neighborTourCost = 406	0.8478	0.6150
Case (ii) neighborTourCost = 260	0.9811	0.8236

It is evident from the Table I that when the neighborTourCost is substantially greater than the bestTourCost (as in case (i)), then its probability of acceptance decreases considerably as compared to the case when the neighborTourCost is moderately greater than the bestTourCost (as in case (ii)). Hence, by modifying the acceptance probability in this way, the tours with the higher cost will have lower probability of acceptance, even though these tours may be close in cost to the current tour. As a result, the search is confined to explore the tours which are close in cost both to the current tour and best tour found so far.

V. EXPERIMENTATION & RESULTS

The basic simulated annealing and the proposed DSA-CE&MAP algorithms have been coded in JavaScript and executed on an Intel core i7 personal computer with clock-speed 3.0 GHz, 8 GB RAM, 4 MB L3 cache via the bash (Ubuntu) command-line interface of Microsoft Windows 10 for some TSPLIB instances. Initial population was generated randomly. Following values were taken for the parameters- (i) initial temperature equals 10^5, (ii) cooling rate equals 0.001. The programs were executed 10 times for each instance. The solution quality is measured by the percentage of excess above the optimal solution value reported in TSPLIB website, as given by the formula.

$$Excess\ (\%) = \frac{Solution\ Value\ -\ Optimum\ Solution}{Optimum\ Solution\ Value} \times 100$$

The tables II and III shows the Excess percentage of best solution values and average solution values over the optimal solution values in 10 runs and the average time of convergence (in second(s)) for each TSPLIB instance . In the tables, the best value, average value and average time is calculated by applying the basic SA and DSA-CE&MAP to the same TSPLIB instance. Furthermore, the excess percentage is calculated as per the above formula in order to compare the solution obtained with the optimal solution. Table II gives the results for fifteen asymmetric TSPLIB instances of size from 17 to 171 and table III gives the results for sixteen symmetric TSPLIB instances of size from 17 to 1379.

TABLE II: Summary of the results by the Basic SA and DSA-CE&MAP for Asymmetric TSPLIB instances

			Basic SA			DSA-CE&MAP		
TSPLIB instance	n	Optimum Value	Best Val(Excess %)	Avg Val(Excess %)	Avg Time(s)	Best Val(Excess %)	Avg Val(Excess %)	Avg Time(s)
br17	17	39	**39 (0.00)**	39.1(0.25)	0.0214	**39(0.00)**	39(0.00)	0.0361
ftv33	34	1286	1657(28.85)	1900.7(47.79)	0.0254	1449(12.67)	1626.8(26.50)	0.3357
ftv35	36	1473	1815(23.21)	2126.2(44.34)	0.0259	1709(16.02)	1904.2(29.27)	0.3819
ftv38	39	1530	2130(39.21)	2398.7(56.77)	0.0269	1757(14.83)	1896.4(23.94)	0.384
p43	43	5620	5639(0.33)	5652(0.56)	0.0247	**5620(0.00)**	5623.4(0.06)	0.3805
ftv44	45	1613	2547(57.90)	2743.3(70.07)	0.0238	1811(12.27)	2295.5(42.31)	0.3404
ftv47	48	1776	2834(59.57)	3182.5(79.19)	0.0286	2380(34.00)	2582(45.38)	0.3443
ry48p	48	14422	16042(11.23)	16697.9(15.78)	0.0383	14853(2.98)	15391.2(6.72)	0.3712
ft53	53	6905	11302(63.67)	13218.7(91.43)	0.0337	9570(38.59)	10781.9(56.14)	0.3601
ftv55	56	1608	2855(77.54)	3242(101.61)	0.0353	2363(46.95)	2638.1(64.06)	0.3609

ftv64	65	1839	3948(114.68)	4286.9(133.11)	0.0406	2761(50.13)	3223.5(75.28)	0.3968
ft70	70	38673	52781(36.48)	54218(40.19)	0.0347	47656(23.22)	49393.4(27.72)	0.474
ftv70	71	1950	4102(110.35)	4569.5(134.33)	0.038	3207(64.46)	3646(86.97)	0.422
kro124p	100	26230	56543(115.56)	61449.2(134.27)	0.0346	43424(65.55)	47762.2(82.08)	0.4572
ftv170	171	2755	13937(405.88)	14622.3(430.75)	0.0487	7578(175.06)	8037(191.72)	6.3129

TABLE III: Summary of the results by the Basic SA and DSA-CE&MAP for Symmetric TSPLIB instances

			Basic SA			DSA-CE&MAP		
TSPLIB instance	n	Optimum Value	Best Val(Excess %)	Avg Val(Excess %)	Avg Time	Best Val(Excess %)	Avg Val(Excess %)	Avg Time(s)
gr17	17	2085	**2085(0.00)**	2088(0.14)	0.0403	**2085(0.00)**	2088(0.14)	0.0374
gr24	24	1272	**1272(0.00)**	1322.5(3.97)	0.0285	**1272(0.00)**	1273.4(0.11)	0.0406
hk48	48	11461	11661(1.74)	12145.4(5.97)	0.0375	**11461(0.00)**	11682(1.92)	0.3545
eil51	51	426	472(10.79)	512.4(20.28)	0.0339	431(1.17)	441(3.52)	0.3994
berlin52	52	7542	8140(7.92)	8369(10.96)	0.0326	**7542(0.00)**	7739.2(2.61)	0.4014
eil76	76	538	737(36.98)	790.8(46.98)	0.039	545(1.30)	562.3(4.51)	0.453
pr76	76	108159	116230(7.46)	120795.4(11.68)	0.0383	108817(0.60)	110474.8(2.14)	0.467
kroA100	100	21282	27253(28.05)	29094.9(36.71)	0.0384	21378(0.45)	22086.3(3.77)	0.5173
kroC100	100	20749	27776(33.8)	29286.7(41.14)	0.0423	20852(0.496)	21585.4(4.03)	0.5212
eil101	101	629	1023(62.63)	1059.4(68.42)	0.335	659(4.76)	670.1(6.53)	0.5046
lin105	105	14379	18983(32.01)	21302.7(48.15)	0.037	14545(1.15)	14874.4(3.44)	0.5319
gil262	262	2378	8067(239.23)	8626.9(262.77)	0.0629	2484(4.45)	2519.2(5.93)	9.928
a280	280	2579	9963(286.31)	10326.6(300.41)	0.0676	2708(5.00)	2760.5(7.03)	10.369
lin318	318	42029	147149(250.11)	153334.2(264.82)	0.0727	43997(4.68)	44589.4(6.09)	11.204
pa561	561	2763	16791(507.70)	17680.3(539.89)	0.1198	3197(15.70)	3256(17.84)	19.568
nrw1379	1379	56638	656923(1059.86)	670889.3(1084.52)	0.400	61309(8.24)	61873.5(9.24)	580.5851

It is observed that the quality of solutions of the algorithms is insensitive to the number of runs. From the tables it is discovered that a greater number of solutions (or tours) with optimum cost can be obtained using DSA-CE&MAP as compared to the basic SA. For example- the asymmetric TSPLIB instances br17 and p43 with optimum values 39 and 5620 respectively, could be solved exactly by DSA-CE&MAP at least once in ten runs, while only br17 with optimum value 39 could be solved exactly by basic SA. Similarly, the symmetric TSPLIB instances gr17, gr24, hk48 and berlin42 with their optimum values 2085, 1272, 11461 and 7542 respectively, could be solved exactly by DSA-CE&MAP at least once in ten runs, while only gr17and gr24 with optimum values 2085 and 1272 respectively, could be solved exactly by basic SA. In addition, the best values and average values for DSA-CE&MAP are better than the basic SA and their corresponding excess percentages are less. Though, basic SA surpasses DSA-CE&MAP in terms of time of convergence, it is noted that, on the basis of the quality of solutions, DSA-CE&MAP outshines basic SA for all the instances, especially for those with the larger number of nodes.

The following figures 1 and 2 depict the graph between the temperature (x-axis) and tour cost (y-axis) for an asymmetric and symmetric TSPLIB instances respectively. It is clearly observed from the figure 1 that DSA-CE&MAP has tour cost (14954) much lower than that of basic SA (16228) and very near to the optimum value (14422). Similarly, it can be identified from the figure 2 that the proposed algorithm has tour cost (21604) much lower than that of basic SA (30674) and very near to the optimum value (21282).

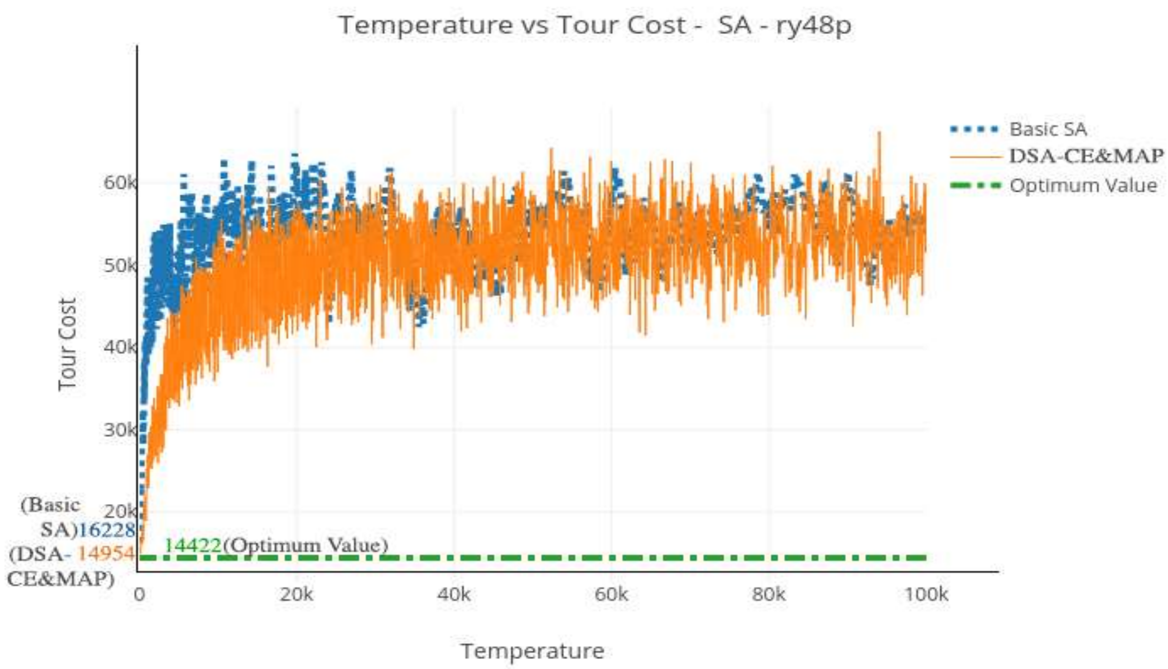

FIGURE 1: Performance of Basic SA and DSA-CE&MAP on Asymmetric TSP instance ry48p(48 nodes)

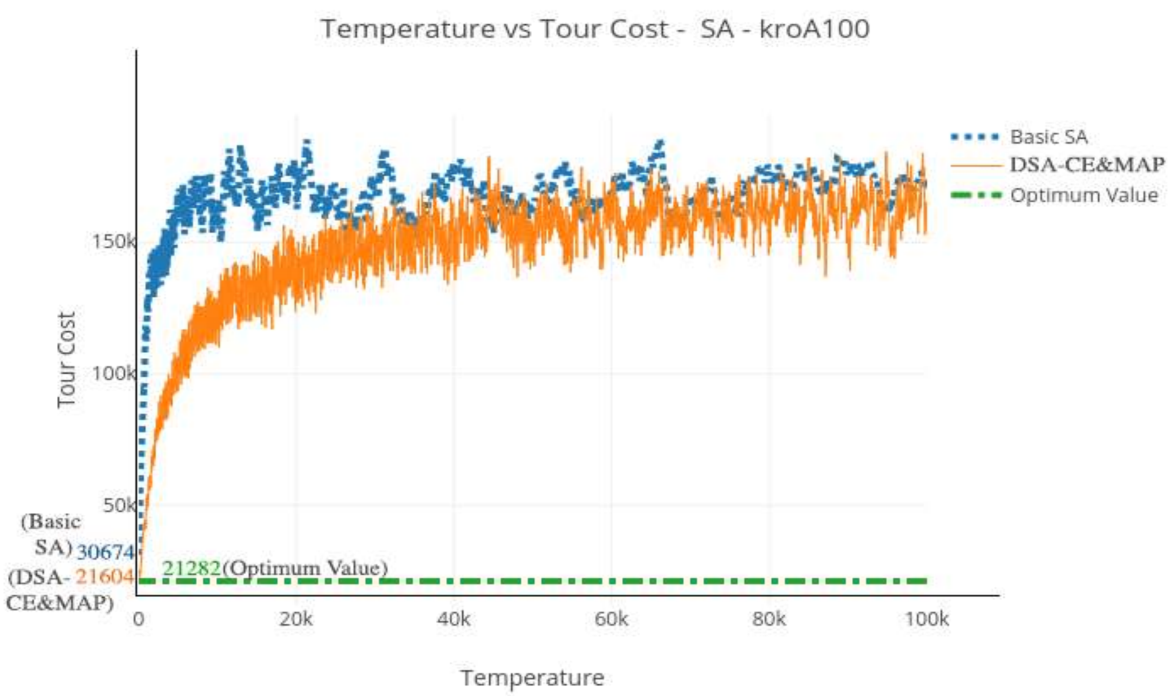

FIGURE 2: Performance of Basic SA and DSA-CE&MAP on Symmetric TSP instance kroA100(100 nodes)

VI. CONCLUSION

In this paper, a refurbished simulated annealing has been proposed by introducing new parameters named "Cooling Enhancer" and "Modified Acceptance Probability" (MAP). It is observed that DSA-CE&MAP provides us with better quality of solutions as compared with the basic SA for all the instances. Moreover, more symmetric and asymmetric TSPLIB instances can be solved exactly using the proposed algorithm. For example- two asymmetric (br17 & p43) and four symmetric (gr17, gr24, hk48 & berlin52) TSPLIB instances could be solved exactly using the proposed approach. On the other hand, only one asymmetric (br17) and two symmetric (gr17 & gr24) TSPLIB instances could be solved exactly using the basic approach. Basic SA outdoes DSA-CE&MAP in respects of time of convergence, since time of convergence is observed to be low for the basic SA.

In future, by making certain changes to the parameters, namely, temperature and cooling rate of the simulated annealing, it is possible to achieve better results and reduce convergence time.

ACKNOWLEDGEMENTS

This research was supported by Department of Computer Engineering, Shri G.S. Institute of Technology & Science, Indore, India.

REFERENCES

[1] TSPLIB instances data source: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html

[2] K.-L. Du and M.N.S. Swamy, "Search and Optimization by Metaheuristics", Springer International Publishing Switzerland,DOI 10.1007/978-3-319-41192-7_2, pp. 29-36, 2016

[3] C.H. Papadimitriou and K. Steiglitz. "Combinatorial Optimization: Algorithms and Complexity". Prentice Hall of India Private Limited, India, 1998.

[4] C.P. Ravikumar. "Solving Large-scale Travelling Salesperson Problems on Parallel Machines". Microprocessors and Microsystems Volume(16) Issue(3), pp. 149-158, 1992.

[5] R.G. Bland and D.F. Shallcross. "Large Travelling Salesman Problems arising from Experiments in X-ray Crystallography: A Preliminary Report on Computation". Operations Research Letters 8,pp. 125-128, 1989.

[6] Rajesh Matai, Surya Singh and Murari Lal Mittal (2010). Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches, Traveling Salesman Problem, Theory and Applications, Prof. Donald Davendra (Ed.), ISBN: 978-953-307-426-9, InTech, pp. 1 - 5.

[7] H. Ghaziri and I. H. Osman, "A neural network algorithm for the traveling salesman problem with backhauls." Computers & Industrial Engineering, Volume(44) Issue(2), pp.267-281, February 2003. http://dx.doi.org/10.106/S0360-8352(02)00179-1

[8] Y. H. Liu, "Different initial solution generators in genetic algorithms for solving probabilistic traveling salesman problem," Applied Mathematics and Computation, Volume(216) Issue(1), pp.125-137, March 2010. http://dx.doi.org/10.106/j.amc.2010.01.021

[9] Y. Liu, S. W. Xiong and H. B. Liu, "Hybrid simulated annealing algorithm based on adaptive cooling schedule for TSP," Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp.895-898, Shanghai, June 12-14, 2009.

[10] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization IEEE Computational Intelligence Magazine," Volume(1) Issue(4), pp.28-39, November 2006.

[11] W. N. Chen, J. Zhang, et al,"A novel set-based particle swarm optimization method for discrete optimization problems," IEEE T. Evolu. Comput., Volume(14) Issue(2), pp.278-300, April 2010. http://dx.doi.org/10.1109/TEVC.2009.2030331

[12] S. Iordache, "Consultant-guided search-a new metaheuristic for combinatorial optimization problems," Proceedings of the 12th annual conference companion on Genetic and evolutionary computation, pp.225-232, July 7-11, Portland, 2010.

[13] Y. Li, A. M. Zhou and G. X. Zhang, "Simulated annealing with probabilistic neighborhood for traveling salesman problems," Proceedings of the Seventh International Conference on Natural Computation, pp.1565-1569, Shanghai, 26-28 July, 2011, http://dx.doi.org/10.1109/ICNC.2011.6022345

[14] B. Liu and P. S. Meng, "Simulated annealing-based ant colony algorithm for traveling salesman problems," Journal of Huazhong University of Science and Technology(Nature Science Edition), Volume(37) Issue(11), pp.26-30 (in Chinese), November 2009.

[15] V. Deineko, B. Klinz and G. Woeginger, "Four point conditions and exponential neighborhoods for symmetric TSP," Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pp.544-553, Miami, January 22-26,2006.

[16] Kirkpatrick S, Gelatt CD Jr, Vecchi MP. Optimization by simulated annealing. Science. New Series, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680.

AUTHORS

First Author: Akshay Vyas, Bachelor of Engineering in Computer Engineering, Shri G.S. Institute of Technology and Science, Indore, India, akshay.vyas95@gmail.com, +91-8602453753
Second Author: Dashmeet Kaur Chawla, Bachelor of Engineering in Computer Engineering, Shri G.S. Institute of Technology and Science, Indore, India, dashmeetkaur11@gmail.com, +91-8827843111
Third Author: Dr. Urjita Thakar, Head, Department of Computer Engineering, Shri G.S. Institute of Technology and Science, Indore, India, uthakar@sgsits.ac.in