Mediterranean Smart Cities Conference (MSCC 2024)
2-4 May, 2024, Martil, Morocco

Global Fuzzy Mamdani Simulated Annealing applied to traveling salesman problem

Hibatallah Meliani
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
hibatallah_meliani@um5.ac.ma

Mohamed Hamza Boulaich
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
mohamedhamza_boulaich@um5.ac.ma

Hanaa El Afia
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
hanaa_elafia@um5.ac.ma

Said Ohamouddou
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
said_ohamouddou@um5.ac.ma

Abdellatif El Afia
National Higher School for Computer Science and Systems Analysis
Mohammed V University in Rabat
Rabat, Morocco
abdellatif.elafia@ensias.um5.ac.ma

Abstract— **The Traveling Salesman Problem (TSP) is an NP-hard combinatorial optimization problem. This study focuses on improving the performance of the simulated annealing (SA) algorithm by automatically adjusting its main parameters through a fuzzy logic system with a Mamdani controller. The performance of this approach is verified on comparison to the traditional SA method in solving the TSP.**

Keywords— Simulated Annealing, Fuzzy logic system, Adaptive parameter control, traveling salesman problem, Mamdani inference system

I. INTRODUCTION

The Traveling Salesman Problem (TSP) is a problem which objective is to find the shortest path among a set of cities, by making sure that each city is visited only once, and ending the path by returning to the start city. The objective is to minimize the total distance traveled by the salesman. TSP can be either symmetric or asymmetric. If symmetric, the distance from city i to city j equal to that from city j to city i. If asymmetric those distances are different. This paper focuses on the symmetric class of TSP.

The main objective of the Traveling Salesman Problem (TSP) is to minimize the overall travel distance. We consider a weighted graph (V, E), where V={1,2,⋯,n} and E are the set of vertices and edges respectively.
TSP mathematical model is as follows:

$$\text{Min C} = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \quad (1)$$

Subject to :

$$\sum_{j=1}^{n} x_{ij} = 1, \forall i \in V \quad (2)$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j \in V \quad (3)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \geq 1, \forall S \subset V, \quad (4)$$

$$x_{ik} \in \{0,1\}, \forall i, j \in V, \quad (5)$$

Where n is the number of cities to visit, dij the distance between city i and city j. Equation (1) represents the objective function that minimizes the traveled distance. Equations (2) and (3) make sure that each city is visited once. Equation (4) eliminate the subtour elimination. In Equation (5), xij is a binary variable that indicates if the salesman travels from city i to city j.

Researchers have developed and used different meta-heuristic algorithms to solve the TSP problem. Some of the metaheuristics used are a JAYA based reinforcement learning algorithm [1], ant colony system [2], simulated annealing [3], List-Based Simulated Annealing [4], hybrid genetic algorithm [5].

To solve the TSP we propose a Global Fuzzy Mamdani Simulated Annealing Algorithm (GFMSA) that uses fuzzy logic mamdani contollers to control the main parameters of SA . In Section 2, we present the algorithms proposed by reseachers to solve TSP. Section 3 explains our proposed algorithm, while Section 4 examines the obtained results from the performance evaluation. The final section concludes the research findings presented in this paper.

II. RELATED WORK

In this work we are going to focus on adapting SA for TSP problem. Many works have used fuzzy logic to control the parameters of SA. In [11], a simulated annealing with adaptive annealing temperature using fuzzy logic system is applied to TSP. In [19], Fuzzy logic with Mamdani inference method is used to control the cooling speed of the simulated annealing algorithm. In [10], the mamdani fuzzy logic controller adapts the SA's neighborhood structure. In [8], fuzzy logic adapts the current temperature and the acceptance probability.In [17], a type2 fuzzy logic was used to control the same parameters as in the previous work.

Other metaheuristics were enhanced using fuzzy logic such as controlling the population size for Ant Colony System (ACS) in [15], the local pheromone in [13] and the pheromone parameters in [2][16].

Other methods such as Hidden Markov Model [25]-[27] was also used to control the parameters of SA. In [20] Hidden Markov Model (HMM) is employed to adapt the cooling schedule of the SA temperature. The same method was used to adapt the neighborhood structure in [18]. The HMM was also used to adapt algorithms applied to TSP such as ACS, where the pheromone parameters, the evaporation parameter and pheromone level exponent were automatically controlled [14] [21]-[24].

To our knowledge, no one has ever controlled all the parameters of the SA using fuzzy logic. In our work, we propose GFMSA a self-adaptive algorithm that uses fuzzy logic with mamdani inference system to adjust the temperature, the acceptance probability and the local search operator of SA to solve TSP.

III. PROPOSED APPROACH

A. Simulated Annealing

Simulated Annealing (SA) are single solution based metaheuristics introduced by Kirkpatrick et al. in 1983. These algorithms improve iteratively a potential solution based on the principles of annealing. A good balance between exploration and exploration has a positive influence on the SA research directions. This balance is achieved by choosing the appropriate parameter setting for temperature, acceptance probability, and local search operator. However, the best parameter settings may change during the process of optimization, which means we need to find rules to update the parameter during the search process and good initial values.

The generalized Simulated Annealing (GSA) [6] is an improved version of SA and a general representation of the classical simulated annealing CSA and the fast simulated annealing FSA. It generates the neigbourhood solution, the probability of acceptance, and the temperature based on predefined functions. Each function has a parameters that changes the characteristics of the algorithm depending on its value.

In this paper we will focus on the temperature and the acceptance probability functions.

The GSA function of the Temperature is defined as:

$$T_{qT}(t) = T_0 \frac{2^{qT-1}-1}{(1+t)^{qT-1}-1} \quad (6)$$

As in [8], for the acceptance probability, we are going to use generalized Metropolis algorithm:

$$p_{qa} = \min\left\{1, [1-(1-q_a)\Delta f\,]^{\frac{1}{1-qa}}\right\} \quad (7)$$

B. Fuzzy Logic System

Lotfi A Zadeh proposed the idea of Fuzzy logic idea in 1965. Fuzzy logic is a mathematical framework used for real world problems that deal with uncertainties. It involves soft or partial truth/false instead or binary crisp true or false.

Fuzzy logic involves key concepts such as fuzzy sets, which allow elements to have degrees of membership. Membership functions, defining the shape of these sets; linguistic variables and terms for human-understandable representations. Fuzzy Logic Operations, introduces operations such as fuzzy AND, fuzzy OR, and fuzzy NOT. Fuzzy rules expressing relationships between inputs and outputs. Fuzzification and defuzzification processes, converting crisp values to fuzzy sets and vice versa. The inference mechanism combines fuzzy rules, determining the degree to which each contributes to the output.

There are different types of fuzzy controller systems, in this paper we are going to use the mamdani fuzzy controllers.

The Mamdani [9] fuzzy controller use linguistic rules in the "IF-THEN" format. These controllers employ fuzzy sets and linguistic variables to represent input and output relationships. The output of a Mamdani controller is determined through fuzzy implication and aggregation methods, resulting in fuzzy sets with degrees of membership. To obtain a crisp output for control actions, a defuzzification step is applied. Mamdani controllers are well suited for systems where heuristic knowledge and linguistic rules play a crucial role in decision-making.

C. Hybridization

In our work, we are going to control the acceptance parameter qa and the temperature parameter qT of the GSA algorithm. The main contribution in our work is controlling all the parameters of GSA including the local search operator used to create new solution for our Traveling salesman problem.

The qa and qT parameters are controlled using the current temperature and fitness difference as input (Figure2), using the rules defined in (Figure1) based on [8].

The membership function of the temperature (Figure3) and acceptance parameters (Figure4), the cost evolution/difference and temperature (Figure2) are defined based on [7] and [8].

qT		*Fitness/Cost difference*		
		Deteriorated	***No Change***	***Improved***
Temperature	High	Fast Decrease	Fast Decrease	Fast Decrease
	Medium	Decrease	Decrease	Slow Decrease
	Low	Decrease	Decrease	Slow Decrease

Figure 1. Rules of the proposed fuzzy system for parameter qT

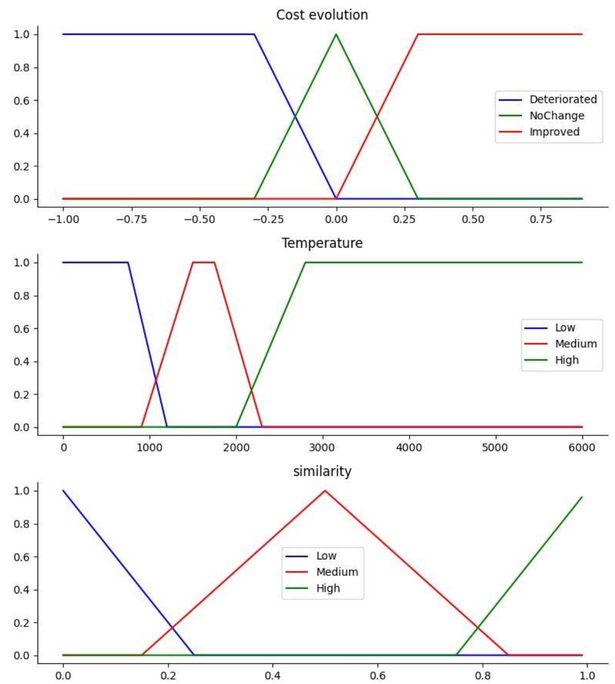

Figure 2. Antecedents Membership function Fuzzy Logic for the fitness difference, current temperature and similarity between solutions

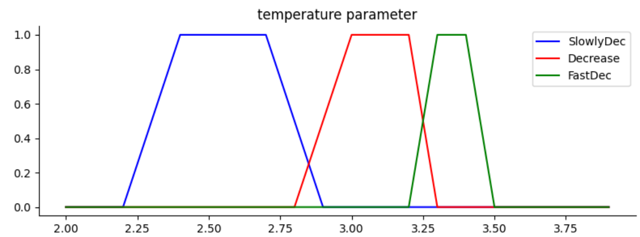

Figure 3. Output membership function for temperature parameter

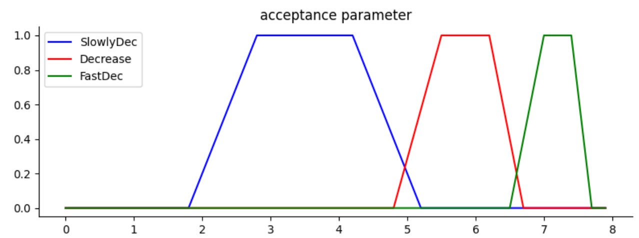

Figure 4. Output membership function for acceptance parameter

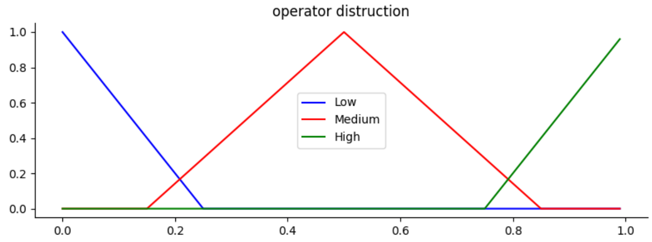

Figure 5. Output membership function for operator destruction parameter

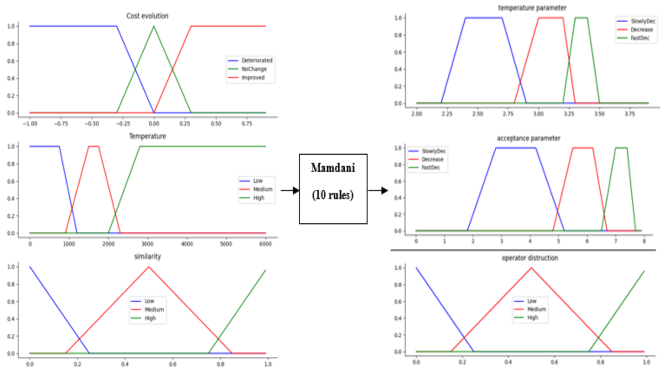

Figure 6. Fuzzy Logic system for SA parameters adaptation with fitness difference, current temperature and similarity between solutions as inputs and temperature parameter, acceptance parameter and operator destruction as output

The local search operators used in our proposed method are swap, insert and scramble. We suppose that those operators has varying impact on the solution. The scramble operator make a higher perturbation to the solution compared to the swap and insert ones.

The fuzzy logic varies the operator depending on the similarity between the current and the past solution, the current temperature and the fitness difference allowing choosing a better local search to create a better solution by either making a small, medium or bigger to the current solution. The fuzzy logic generate a value called operator destruction which value is between 0 and 1. If the value of the operator destruction is greater than 0.5 the proposed algorithm applies the scramble operator, if between 0.5 and 0.25 the insert operator is used else we chose the swap one.

The similarity between solution is calculated using similarEdgesPercentage() in Algorithm1. It calculates the percentage of similar edges between the current and the new solution. The generateSolution() is the function that generates a new solution based on the previous solution, depending on the operator destruction value controlled by the fuzzy logic system.

Algorithm1: The proposed GFMSA algorithm
Input: Objective functions f, Maximum number of iteration($MaxIter$), initial temperature T_init **Output:** best solution for the objective functions f Initialize parameters T_init, qa (acceptance parameter), qt (temperature parameter), op (operator_destruction_value) Ti ←T_init S ← initialSolution() $Sbest \leftarrow S$ **for** i ← 1 to $MaxIter$ do S' ← **generateSolution**(op, S) Δf ← f(S') – f(S) normΔf ← normalize(Δf) similarity ← **similarEdgesPercentage** (S', S) **if** $\Delta f < 0$ or (min {1, $[1 - (1 - qa)\Delta f]^{1/(1-qa)}$} > $random$[0,1] S ← S' **end** **if** f(S') < f($Sbest$): $Sbest$ ← S' **end** qt, qa, op←**fuzzyLogicController**(Ti, normΔf , similarity) $Ti \leftarrow T\_init \frac{2^{qt-1}-1}{(1+i)^{qt-1}-1}$ **end**

		GFMSA	*SA*	*p-value*
rd100	Min Avg CPU	**44692** **48196** 5.13	44980 48264 0.26	0.85
att48	Min Avg CPU	113031 123951 3.63	**86831** **105116** 0.12	0.000
a280	Min Avg CPU	**2818** **2818** 7.88	30366 31697 1.08	0.000
ch150	Min Avg CPU	**45358** **48345** 6	46006 48584 1	0.046
d198	Min Avg CPU	**22514** **22514** 7.07	148724 161647 0.64	0.000
eil101	Min Avg CPU	**2008** **2139** 5.12	2866 3051 0.28	0.000
d657	Min Avg CPU	**232140** **232140** 14	760568 793613 2.13	0.000

TABLE I. PERFORMANCE OF THE SA COMPARED TO GFMSA ON TSP INSTANCES

IV. PERORMANCE EVALUATION

To study the behaviour of our improved GFMSA, we tested it on several TSP benchmark instances. The proposed algorithm is compared with classical simulated annealing annealing (SA) with a cooling rate of 0.9999978 used in [12]. For both algorithms, the maximum iteration is 1000, the initial temperature is 5000. The algorithms were tested on every instance 100 times using python language, SciKit-Fuzzy library and run on a PC with Intel Core i5 2.30GHz processor and 4 GB of RAM.

The results in Table1 show that GFMSA gives better solution compared to SA, which means that the fuzzy logic controller helps the SA search for better solution by controlling the local search operator type, the probability of the acceptance of the new generated solution and the cooling schedule. The proposed algorithm takes more computation time due to the added steps needed in the mamdani fuzzy logic controller.

The non-parametric statistical test Wilcoxon signed-rank test with a significant level of 0.05 is used to calculate the p-values of the results in each instance of the dataset.Table1 shows that the p-values obtained are less than 0.05 which mean that the difference obtain in the results between the algorithms tested are significant. We can say that proposed algorithm outperform the standard SA.

V. CONCLUSION

The primary focus of this paper is to assess the efficacy of fuzzy simulated annealing in addressing the Traveling Salesman Problem (TSP). A comparative analysis has been conducted between the conventional simulated annealing and its fuzzy-controlled across various instances from benchmark datasets. The results demonstrate the effectiveness of simulated annealing, with the fuzzy controller notably enhancing the algorithm's performance for the specified TSP instances. Future research directions include the exploration of alternative TSP versions using the Global Fuzzy Mamdani Simulated Annealing (GFMSA). Additionally, extending the study to evaluate GFMSA performance on larger TSP datasets and comparing outcomes with other metaheuristic approaches.

REFERENCES

[1] Xu J, Hu W, Gu W, Yu Y. A Discrete JAYA Algorithm Based on Reinforcement Learning and Simulated Annealing for the Traveling Salesman Problem. Mathematics. 2023; 11(14):3221.

[2] Bouzbita, S., El Afia, A., and Faizi, R. (2020). The behaviour of ACS-TSP algorithm when adapting both pheromone parameters using fuzzy logic controller. International Journal of Electrical and Computer Engineering, 10(5), 5436. DOI: 10.11591/ijece.v10i5.

[3] Liu, X., Zhang, B., & Du, F. (2014). Integrating Relative Coordinates with Simulated Annealing to Solve a Traveling Salesman Problem. 2014 Seventh International Joint Conference on Computational Sciences and Optimization.

[4] Shi-hua Zhan, Juan Lin, Ze-jun Zhang, Yi-wen Zhong, "List-Based Simulated Annealing Algorithm for Traveling Salesman Problem", Computational Intelligence and Neuroscience, vol. 2016, Article ID 1712630, 12 pages, 2016.

[5] Vladimir Ilin, Dragan Simić, Svetislav D Simić, Svetlana Simić, Nenad Saulić, José Luis Calvo-Rolle, A hybrid genetic algorithm, list-based simulated annealing algorithm, and different heuristic algorithms for the travelling salesman problem, Logic Journal of the IGPL, Volume 31, Issue 4, August 2023, Pages 602–617

[6] Tsallis, C., & Stariolo, D. A. (1996). Generalized simulated annealing. Physica A: Statistical Mechanics and Its Applications, 233(1-2), 395–406.M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.

[7] Agostini, F. P., Soares-Pinto, D. D. O., Moret, M. A., Osthoff, C., & Pascutti, P. G. (2006). Generalized simulated annealing applied to protein folding studies. Journal of Computational Chemistry, 27(11), 1142–1155.

[8] Lalaoui, M., & El Afia, A. E. (2018). A Fuzzy generalized simulated annealing for a simple assembly line balancing problem. IFAC-PapersOnLine, 51(32), 600–605.

[9] Mamdani, E. H. (1976). Advances in the linguistic synthesis of fuzzy controllers. International Journal of Man-Machine Studies, 8(6), 669–678.

[10] Lalaoui, M., El Afia, A., & Chiheb, R. (2018). Simulated Annealing with Adaptive Neighborhood using Fuzzy Logic Controller. Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications - LOPAL '18

[11] Qin, L., Wang, J., Li, H., Sun, Y., & Li, S. (2017). An Approach to Improve the Performance of Simulated Annealing Algorithm Utilizing the Variable Universe Adaptive Fuzzy Logic System. IEEE Access, 5, 18155–18165.

[12] Wang, Z., Geng, X., & Shao, Z. (2009). An Effective Simulated Annealing Algorithm for Solving the Traveling Salesman Problem. Journal of Computational and Theoretical Nanoscience, 6(7), 1680–1686.

[13] El Afia, A., Bouzbita, S., and Faizi, R. (2017). The effect of updating the local pheromone on acs performance using fuzzy logic. International Journal of Electrical and Computer Engineering, 7(4), 2161. DOI: 10.11591/ijece.v7i3

[14] Bouzbita, S., El Afia, A., and Faizi, R. (2018, May). Parameter adaptation for ant colony system algorithm using Hidden Markov Model for TSP problems. In Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications (pp. 1-6). DOI:10.1145/3230905.3230962

[15] Bouzbita, S., El Afia, A., and Faizi, R. (2019, September). Adjusting Population Size of Ant Colony System Using Fuzzy Logic Controller. In International Conference on Computational Collective Intelligence (pp. 309-320). Springer, Cham. DOI: 10.1007/978-3-030-28374-2_27

[16] Bouzbita, S., El Afia, A. (2022). A Fuzzy Meta Model for Adjusting Ant Colony System Parameters. In book: Proceedings of the 5th International Conference on Big Data and Internet of Things (pp.48-58). DOI:10.1007/978-3-031-07969-6_4

[17] Lalaoui M., A. El Afia, (2018), A Versatile Generalized Simulated Annealing Using Type-2 Fuzzy Controller for the Mixed-Model Assembly Line Balancing Problem, the 9th IFAC Conference on Manufacturing Modelling, Management and Control, Berlin, Germany, August 28-30, 2019. doi: 10.1016/j.ifacol.2019.11.633.

[18] Lalaoui M., El Afia A., Chiheb R. (2021). Dynamic Simulated Annealing with Adaptive Neighborhood Using Hidden Markov Model. In: Yalaoui F., Amodeo L., Talbi EG. (eds) Heuristics for Optimization and Learning. Studies in Computational Intelligence, vol 906. Springer, Cham. Doi: 10.1007/978-3-030-58930-1_11.

[19] Lalaoui M., A. El Afia, R. Chiheb, (2017). Fuzzy logic controller for an adaptive Huang cooling of simulated annealing, The 2nd International Conference on Big Data, Cloud and Applications (CloudTech'17) IEEE Conference, Rabat, Morocco. https://doi.org/10.1145/3090354.3090420

[20] Lalaoui M., A. El Afia and R. Chiheb, (2016). Hidden Markov Model for a Self-Learning of Simulated Annealing Cooling Law, the 5th International Conference on Multimedia Computing and Systems IEEE Conference, ICMCS'16, Marrakech, Morocco. doi: 10.1109/ICMCS.2016.7905557

[21] Bouzbita, S., El Afia, A., Faizi, R., and Zbakh, M. (2016, May). Dynamic adaptation of the ACS-TSP local pheromone decay parameter based on the Hidden Markov Model. In 2016 2nd international conference on cloud computing technologies and applications (CloudTech) (pp. 344-349). IEEE. DOI: 10.1109/CloudTech.2016.7847719

[22] Bouzbita, S., El Afia, A., and Faizi, R. (2016). A novel based Hidden Markov Model approach for controlling the ACS-TSP evaporation parameter. In 2016 5th international conference on multimedia computing and systems (ICMCS) (pp. 633-638). IEEE. DOI: 10.1109/ICMCS.2016.7905544

[23] Bouzbita, S., El Afia, A., and Faizi, R. (2019). Hidden markov model classifier for the adaptive ACS-TSP pheromone parameters. In bioinspired Heuristics for Optimization (pp. 153-169). Springer, Cham. DOI: 10.1007/978-3-319-95104-1_10

[24] Bouzbita, S., El Afia, A., and Faizi, R. A New Hidden Markov Model Approach for Pheromone Level Exponent Adaptation in Ant Colony System. In Heuristics for Optimization and Learning (pp. 253-267). Springer, Cham. DOI: 10.1007/978-3-030-58930-1_17

[25] Lalaoui M., A. El Afia, R. Chiheb.(2017). A Self-Tuned Simulated Annealing Algorithm Using Hidden Markov Model, International Journal of Electrical and Computer Engineering (IJECE). doi: 10.11591/ijece.v8i1.pp291-298

[26] Afia, A. E., Lalaoui, M., & Chiheb, R. (2019). A Self Controlled Simulated Annealing Algorithm using Hidden Markov Model State Classification. Procedia Computer Science, 148, 512–521. doi:10.1016/j.procs.2019.01.024

[27] Lalaoui, M., Afia, A. E., & Chiheb, R. (2017). A self-adaptive very fast simulated annealing based on Hidden Markov model. 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech). doi:10.1109/cloudtech.2017.8284698