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

ROBOT PATH PLANNING USING FUZZY TSUKAMOTO SIMULATED ANNEALING

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

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

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— **In this paper we present an adaptive Simulated Annealing (SA) algorithm designed to solve the path planning problem. The fuzzy logic based on Tsukamoto controller, automatically controls the acceptance probability, the temperature cooling schedule and the local search operator of the SA algorithm. The adaption of this parameters is based on the current temperature, the fitness improvement and the solutions similarity. The performance of the proposed algorithm is verified thought a comparison with the traditional SA on benchmark datasets for path planning problem.**

Keywords— Simulated Annealing, Fuzzy logic system, parameter control, Robot path planning problem, Tsukamoto controller

I. INTRODUCTION

The path planning problem goal is to allow the robot to find a collusion free, optimal path that start from a chosen point to an end point in a specified environment.

The path planning problem is used in different fields, such as game development, artificial intelligence and traffic route navigation. This problem has two types. The first one assumes that the environment is static and in the second one, the environments and the obstacles are dynamic.

In this paper, we are assuming that the environment is known and static and the function to optimize is the path length. The environment where the robot is navigating is a 2D map that includes a finite set of feasible and non-feasible points:

The fitness function of a path $P = \{p_1, p_2, ..., p_{n-1}, p_n\}$ is:

$$Fitness(P) = \sum_{i=1}^{n-1} d_{i,i+1} \quad (1)$$

Where $d_{i,i+1}$ is the Euclidean distance between point $p_i = (x_i, y_i)$ and $p_{i+1} = (x_{i+1}, y_{i+1})$. This distance formula is:

$$d_{i,i+1} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (2)$$

The Robot Path Planning Problem (RPPP) was solving using many methods [35] such as ant colony algorithm in [30], Particle Swarm Optimization [31], Dijkstra's algorithm [32], neural network-based approach [33], genetic algorithm [34].

In this paper, we are going to present our Fuzzy Tsukamoto Simulated Annealing (FTSA) algorithm that adaptively control the main parameters of SA depending on the optimization process. In Section 2, we are going to present the research related to our work. In Section 3, we are going to describe our proposed algorithm. Section 4 carries on the experimental methodology. The last section concludes the research results of this paper.

II. RELATED WORK

In this work, we are going to use a self-adaptive algorithm based on simulated annealing (SA) and Fuzzy Logic system to solve the Robot Path Planning Problem (RPPP). SA is a metaheuristic algorithm inspired by the principles of annealing. The paper [2] applied the SA with adaptive cooling and used three different representations for the path: Bésier, polyline and spline interpolated curve to solve the RPPP. The paper highlighted the importance of calibrating the weight related to the number of points of the path, its length, and obstacles collusion. In [3], SA was incorporated into ant colony process to improve its performance and an entropy increase strategy was added to avoid the premature convergence and was applied to the path planning problem. In [4], SA and a fuzzy logic were used to solve navigation problem for autonomous mobile robot with fixed obstacles. The SA generates feasible paths and fuzzy logic is used for trajectory tracking.

Concerning the control of parameters. In [8] and [20] the cooling schedule and the acceptance of the Generalized Simulated Annealing are adapted using Fuzzy logic. In [22], Mamdani controllers were used to control the cooling speed of the simulated annealing algorithm. In [10], the neighborhood structure was controlled using the same method. In [16]-[19] and [23] fuzzy logic was also used to adapt parameters of other metaheuristics such as the Ant Colony System algorithm (ACS).

Other methods such as Hidden Markov Model (HMM) [24] were also used to adapt parameters of SA [36] [37]. In [25] HMM controls the cooling schedule of the SA temperature and the neighborhood structure in [21]. HMM was also able to successfully adjust parameters of population based metaheuristics such as ACS [17] [26]-[29].

To our knowledge, we are the first to adapt all the parameters of the SA using fuzzy logic with Tsukamoto inference system. In this paper, we propose FTSA a self-adaptive algorithm that uses Tsukamoto fuzzy inference system to adjust the acceptance probability, temperature cooling schedule, and the local search operator of SA.

III. THE PROPOSED ALGORITHM

A. *Simulated Annealing*

Simulated Annealing (SA) is a metaheuristic inspired by the annealing process of metals. Kirkpatrick et al. [5] introduced the Algorithm in 1983. The algorithms tries to find the best solution for an optimization problem by balancing exploration and exploitation using the annealing process. The algorithm start with high temperature allowing the discovery for non-optimal solution then the algorithm start reducing the temperature and conserving area of good solutions.

Therefore, a good balance between exploration and exploration improves the performance of SA, which is achieved through a selection of good parameters for temperature, temperature cooling schedule, acceptance probability, and local search operator. The best parameter settings are different during each process of optimization, which means an online parameter selection is necessary to control the trade of between exploration and exploitation.

The Generalized Simulated Annealing (GSA) [6] has two parameters: qv that represents the visiting parameter and qa called the acceptance parameter. When qv/2 = qa =- 1 the algorithm represent the fast simulated annealing when qv = qa = 1 the algorithm acts as a classical simulated annealing. The cooling schedule, the probability of acceptance and the neighborhood structure are represented by mathematical equations using the qv and qa parameters.

Since we are only going to focus in this paper on the temperature and the acceptance probability functions of GSA, we changed qv in the temperature equation to qT and we call it the temperature parameter.

The GSA function of the temperature is:

$$T_{qT}(t) = T_0 \frac{2^{qT-1}-1}{(1+t)^{qT-1}-1} \quad (3)$$

We are going to use the use generalized Metropolis algorithm for the acceptance probability as in [8]:

$$p_{qa} = \min\left\{1, [1-(1-q_a)\Delta f]^{\frac{1}{1-qa}}\right\} (4)$$

Where qa is the acceptance parameter and qT is the temperature parameter, T_0 is the initial temperature and Δf is the fitness difference.

Our goal is to use Fuzzy Logic system based on Tsukamoto inference system to improve the GSA by adapting qa and qT and the local search operator.

B. *Fuzzy Logic System*

In 1965, Lotfi A. Zadeh introduced the concept of Fuzzy Logic, a mathematical framework designed for addressing real-world problems characterized by uncertainties. This approach involves key elements such as fuzzy sets, allowing for degrees of membership, membership functions that define set shapes, and linguistic variables with terms for human-understandable representation. Fuzzy logic operations, including fuzzy AND, fuzzy OR, and fuzzy NOT, play a crucial role, along with fuzzy rules that articulate relationships between inputs and outputs. Fuzzification and defuzzification processes facilitate the conversion of crisp values to fuzzy sets and vice versa. The inference mechanism integrates fuzzy rules, assessing the extent to which each rule influences the final output.

Fuzzy control systems leverage these principles to model and control processes, demonstrating adaptability and learning capabilities.

There are various types of fuzzy controller systems; this paper focuses specifically on Tsukamoto fuzzy controllers [9]. Operating on the foundation of linguistic rules expressed in the "IF-THEN" format, Tsukamoto controllers employ fuzzy sets and linguistic variables to model the relationships between input and output variables. In this approach, each fuzzy rule's result is represented through a fuzzy set characterized by a monotonic membership function. Tsukamoto controllers opt for a weighted average approach. In Tsukamoto defuzzification, each rule's output contributes to the final crisp output based on its firing strength.

C. *SA PARAMETER CONTROL USING FUZZY LOGIC*

The performance of the SA algorithm heavily depends on the assigned values to its parameters, influencing its convergence speed. The key contribution of our work lies in the adaptive control of all significant parameters of the GSA, including the local search operator employed to generate new solutions for our robot path-planning problem using a fuzzy logic system based on Tsukamoto inference system with nine rules (Figure5).

The fuzzy logic varies the operator depending on the current temperature and the fitness difference (Figure1) allowing choosing a better local search to create a better solution by either making a small or bigger change to the current solution. Therefore, we choose either to make mutation operator where a single node of the path is replace with a new feasible node from the environment map of our RPPP or to create a completely new solution. The qA and qT parameters are controlled also using the current temperature and fitness difference as input.

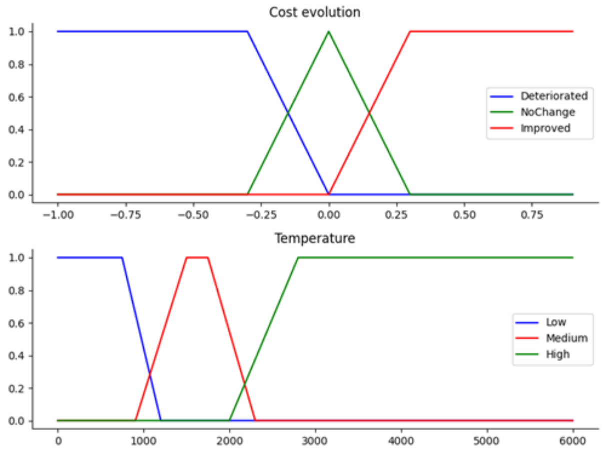

Figure 1. Membership function for the inputs temperature and fitness difference

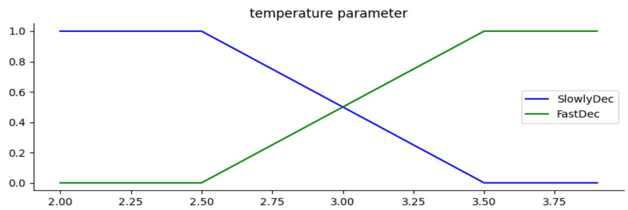

Figure 2. Output membership function for temperature parameter

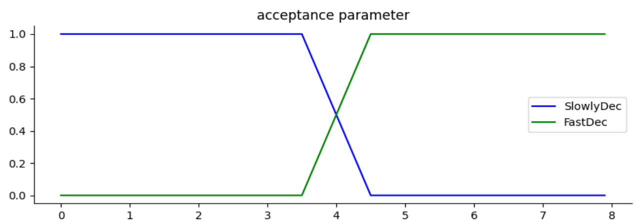

Figure 3. Output membership function for acceptance parameter

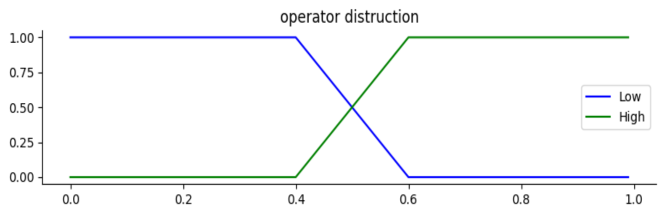

Figure 4. Output membership function for operator destruction parameter

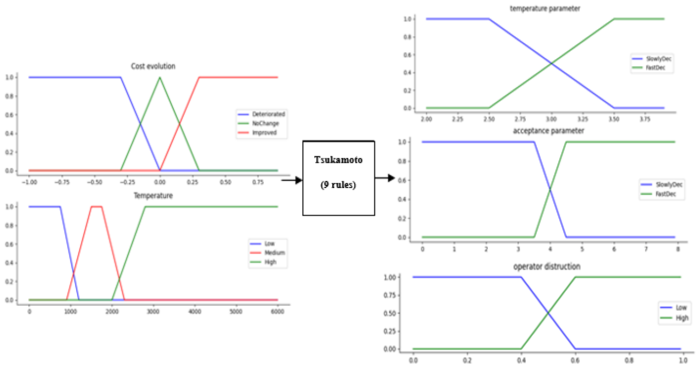

Figure 5. Fuzzy Logic system for SA parameters adaptation with fitness difference and current temperature as inputs

In generateSolution() function in Algorithm1, generate a new solution based on the previous solution, depending on the operator destruction value controlled by the fuzzy logic system. If the value is low, the algorithm execute a mutation on the previous solution. If the value if high a new solution is generated using the initialization method proposed in [13]. The fuzzyLogicController function takes as imput the current temperature and the normalized fitness difference and return the temperature parameter, the acceptance parameter used to update the temperature and the acceptance probability respectively and the operator destruction value used to create the new solution for the SA algorithm.

Algorithm1: The proposed FTSA 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)

$Ti \leftarrow T$_init
S ← initialSolution()
$Sbest \leftarrow S$

for i ← 1 to $MaxIter$ do
  S' ← **generateSolution**(op, S)
  Δf ← f(S') – f(S)
  normΔf ← normalize(Δf)
  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)
  $Ti \leftarrow T\_init \frac{2^{qt-1}-1}{(1+i)^{qt-1}-1}$

end

IV. PERORMANCE EVALUATION

To study the behavior of our improved FTSA, we tested it on several Path planning benchmark instances from movingai.com [14] [15] and compare it to classical SA. As stated in reference [12], according to [11], a larger initial temperature and a smaller cooling rate increase the likelihood of discovering the optimal solution in SA for robot path planning. However, this choice also necessitates a longer processing time for execution. So we will be using the maximum temperature defined in our fuzzy logic system in both FTSA and SA. The cooling rate of SA will be 0.9999978. For both algorithms, the maximum iteration is 100. The algorithms were tested on every instance 30 times using Python language and SciKit-Fuzzy library and run on a PC with Intel Core i5 2.30GHz processor and 4 GB of RAM.

While testing our FTSA and SA on the chosen benchmark on different instances, the results were the same in term of optimal solution. Both algorithm struggle in some instances to find the initial path because our choice of the method used to initialize the first solution wasn't optimal for some type of instance. Therefore, we choose to compare the algorithms to instances where the SA struggle to find the optimal solution from the first run. The results in Table1 show that FTSA gives better solution compared to SA in the instances where both algorithm struggle to find the best solution from the first run. This 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 has more computation time due to the added steps needed in the tsukamoto fuzzy logic controller used for the automatic adjustment of the parameters.

Significance of the results is checked using the Wilcoxon signed-rank with 0.05 level. The p-values shown in Table1 are greater than 0.05, which mean that the results of SA and FTSA are not showing significant difference. Meanwhile, we observe a large effect size, which indicates a practically significant difference in performance between the two algorithms. This suggests that the observed differences, while meaningful, are not statistically significant with the current sample size.

TABLE I. PERFORMANCE OF THE SA COMPARED TO FTSA ON RPP INSTANCES

		FTSA	*SA*	*p-value*	*Effect size*
80	Min Avg CPU	**31.63** **53.56** 0.49	34.06 34.06 1.15	0.39305	34.68
82	Min Avg CPU	**33.30** **48.20** 1.06	33.35 58.40 1.26	0.069893	26.29
84	Min Avg CPU	**31.84** **46.91** 0.50	33.06 52.47 0.21	0.41613	35.05
86	Min Avg CPU	**32.31** **51.60** 0.23	36.11 58.93 0.53	0.177193	30.30
88	Min Avg CPU	**34.11** **54.86** 0.50	37.87 61.31 0.26	0.542528	36.87
89	Min Avg CPU	**32.65** 54.86 2.98	34.27 **50.03** 0.16	0.427955	35.23

V. CONCLUSION

This paper primarily aims to evaluate the effectiveness of fuzzy simulated annealing in solving the Robot Path Planning problem. The proposed algorithm was compared to classical SA using benchmark datasets. The findings highlight the efficacy of simulated annealing, particularly showcasing the performance of the fuzzy Tsukamoto controller in the specified scenarios. Future research directions involve exploring alternative methods for generating feasible paths. Additionally, there are plans to extend the study by assessing the performance of FTSA on larger datasets and samples and comparing the results with other metaheuristic approaches.

REFERENCES

[1] Yun-Qian Miao, Alaa M. Khamis, Fakhri Karray, and Mohamed S. Kamel , A NOVEL APPROACH TO PATH PLANNING FOR AUTONOMOUS MOBILE ROBOTS, Control and Intelligent Systems, Vol. 39, No. 4, 2011

[2] Marcos de Sales Guerra Tsuzuki, Thiago de Castro Martins, Fabio Kawaoka Takase, ROBOT PATH PLANNING USING SIMULATED ANNEALING, INCOM'2006: 12th IFAC/IFIP/IFORS/IEEE/IMS Symposium,

[3] Kun Liu and Mengxia Zhang, Path planning based on simulated annealing ant colony algorithm, 2016 9th International Symposium on Computational Intelligence and Design

[4] Horacio Martınez-Alfaro, Simon Gomez-Garcıa, Mobile robot path planning and tracking using simulated annealing and fuzzy logic control, Expert Systems with Applications 15 (1998) 421–429.

[5] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, Optimization by Simulated Annealing, Science, 13 May 1983, Volume 220, Number 4598

[6] Tsallis, C., & Stariolo, D. A. (1996). Generalized simulated annealing. Physica A: Statistical Mechanics and Its Applications, 233(1-2), 395–406.

[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] Tsukamoto, Yahachiro, and Yahachiro Tsukamoto. AN APPROACH TO FUZZY REASONING METHOD. Jan. 1993,

[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] Q. Zhu, Y. Yan, and Z. Xing, "Robot path planning based on artificial potential field approach with simulated annealing," in Proc. Sixth Int. Conf. on Intelligent System Design & Applications, vol. 2. IEEE, 2006, pp. 622–627.

[12] Miao, H., & Tian, Y.-C. (2008). Robot path planning in dynamic environments using a simulated annealing based approach. 2008 10th International Conference on Control, Automation, Robotics and Vision. doi:10.1109/icarcv.2008.4795701

[13] Lee, J., & Kim, D.-W. (2016). An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph. Information Sciences, 332, 1–18. doi:10.1016/j.ins.2015.11.004

[14] Sturtevant, N. R. (2012). Benchmarks for grid-based pathfinding. IEEE Transactions on Computational Intelligence and AI in Games, 4(2), 144-148. DOI: 10.1109/TCIAIG.2012.2197681

[15] https://www.movingai.com/benchmarks/dao/index.html

[16] 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

[17] 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

[18] 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

[19] 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.

[20] 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.

[21] 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.

[22] 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. doi: 10.1145/3090354.3090420

[23] 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

[24] 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

[25] 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

[26] 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

[27] 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

[28] 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

[29] 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

[30] Chen, Y., & Zhou, X. (2021). Path planning of robot based on improved ant colony algorithm in computer technology. Journal of Physics: Conference Series IOP Publishing, 1744(4), Article 042092.

[31] Fei W., Ziwei W., Meijin L., Robot Path Planning Based on Improved Particle Swarm Optimization. 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). IEEE, 2021: 887-891.

[32] Fink, W., Baker, V. R., Brooks, A. J. W., et al. (2019). Globally optimal rover traverse planning in 3D using Dijkstra’s algorithm for multi-objective deployment scenarios. Planetary and Space Science, 179, Article 104707.

[33] Godio, S., Primatesta, S., Guglieri, G., et al. (2021). A Bioinspired neural network-based approach for cooperative coverage planning of UAVs. Information, 12(2), 51.

[34] Tao, Y., Wen, Y., Gao, H., et al. (2022). A Path-Planning Method for Wall Surface Inspection Robot Based on Improved Genetic Algorithm. Electronics, 11(8), 1192.

[35] Lixing Liu, Xu Wang, Xin Yang, Hongjie Liu, Jianping Li, Pengfei Wang, Path planning techniques for mobile robots: Review and prospect, Expert Systems with Applications, Volume 227, 2023, 120254, ISSN 0957-4174, doi: 10.1016/j.eswa.2023.120254.

[36] 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

[37] 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