International Journal of Engineering & Technology, 9 (4) (2020) 857-862

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Path Planning for Indoor UAV Using A^* and Late Acceptance Hill Climbing Algorithms Utilizing Probabilistic Roadmap

Jacob Hopkins[1], Forrest Joy[1], Alaa Sheta[2*], Hamza Turabieh[3] and Dulal Kar[1]

[1]*Dept. of Computing Sciences, Texas A&M University-Corpus Christi, Corpus Christi, TX 78412, USA*
[2]*Computer Science Department, Southern Connecticut State University, New Haven, CT 06515*
[3]*Department of Information Technology, Taif University, Taif, Saudi Arabia*

Abstract

The main objective of an unmanned aerial vehicle (UAV) path planning is to generate a flight path that links a start point to an endpoint in an indoor space avoiding obstacles. Path planning is essential for many real-life applications such as an autonomous car, surveillance mission, farming robots, unmanned aerial vehicles package delivery, space exploration, and many others. To create an optimal path, we need to adopt a specific criterion to minimize the distance the UAV must travel such as the Euclidean distance. In this paper, we provide our initial idea of creating an optimal path for indoor UAV using both A^* and the Late Acceptance Hill Climbing (LAHC) algorithms. We are adopting an indoor search environment with various complexity and utilize the Probabilistic Roadmap algorithm (PRM) as a search space for both algorithms. The basic idea following PRM is to generate random sample points in the space and search these points for an optimal path. The developed results show that the LAHC algorithm outperforms the A^* algorithm.

Keywords: *UAV; Probabilistic Road Mapping, A^*, Late Acceptance Hill-Climbing.*

1. Introduction

Recently, UAVs became one of the most challenging and elevated technologies in aeronautics [1]. UAV shows many advantages in various military, and civilian applications such as low power-driven, unmanned, excellent concealment, low cost, and exceptional maneuverability. UAV has been utilized for commercial, scientific, precision agriculture [2,3], surveillance, product deliveries, aerial photography, earth resources monitoring [4], and border security [5].

Some statistics show that UAV sales in Germany touched 400,000 units in 2017 and are expected to reach one million in 2020 [6]. Likewise, the National Purchase Diary Panel (NPD) located in the USA estimated that UAV sales are doubled in 2017. The unmanned aircraft system (UAS) was coined by the United States Department of Defense (DoD) and the United States Federal Aviation Administration in 2005 corresponding to the Unmanned Aircraft System Roadmap 2005–2030. Although the use of UAV has been massively used in outdoor applications, the latest technology facilitated the use of UAV for indoor navigation applications. In Figure 1, we show several types of drones that are used for many indoor applications. Flying an indoor drone can come for several reasons such as practice, racing, and professional applications. Professional indoor drones are being used for inspections, security, and 3D modeling of an environment.

Solving a UAV path planning problem depends on searching for an optimal collision-free path from a given launch node to a target in an obstacle environment adopting a certain evaluation criterion. Many research articles explored the use of UAV for mission planning in an indoor environment with a set of assumptions based on the characteristics of the indoor environment. For example, MIT's Robust Robotics Group developed a path planning with obstacles in an indoor environment using the Belief Roadmap (BRM) algorithm that integrate a predictive model to sense the environment [7]. Many techniques were proposed to develop an optimal UAV path for various missions. In [8], the author provided a new algorithm for collision-free path planning using an ant colony optimization (ACO) algorithm taking into consideration both dynamic threats and static obstacles.

Probabilistic Road Mapping is known as a motion planning algorithm that is commonly used in robotics control for path planning and obstacle avoidance. The PRM algorithm has many advantages since it can work in various environment and the algorithm is pretty fast with satisfactory complexity. In the past, several hybrid algorithms were introduced such as PRM-GA [9], ACO-PRM [10], and Potential Field-ACO [11]. An example of a PRM is shown in Figure 2.

In this research, we provide our initial idea of using A^* and

(a) Elios 2 autonomous inspection drone. **(b)** The DJI Mavic 2 Pro. **(c)** FPV Racing Indoor Drone.

Figure 1: Several types of drone for indoor applications.

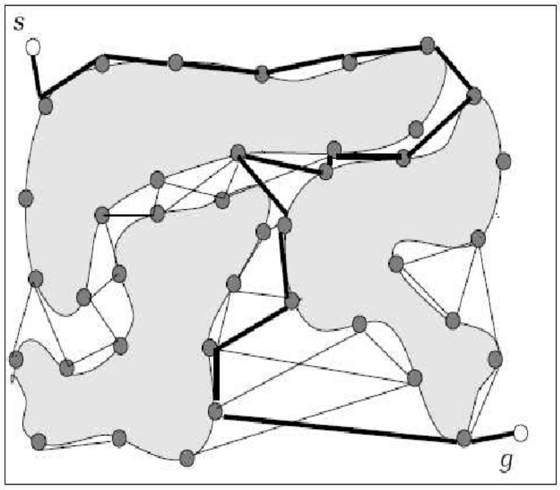

Figure 2: Example of an obstacle-based probabilistic roadmap. s and g are the starts and goal configurations. A path is found by searching over the roadmap. The image was taken from [12].

Late Acceptance Hill Climbing to develop a collision path planning algorithm based on the Probabilistic Roadmap. This research provides a framework that can be extended to various search algorithms. This paper is organized as follows. In Section 3, we provide an overview of the research developed in the area of path planning using both classical and heuristics methods. The basic principles of the PRM are presented in Section 2. Section 3 explores the importance of the path planning problem and how it can be solved using heuristic and meta-heuristic algorithms. The obtained results for A^* and LAHC algorithms are presented in Section 4. Finally, the conclusion and future works are presented in Section 5.

2. Probabilistic Roadmap

In the last few decades, path planning algorithms based Probabilistic Roadmap algorithm (PRM) became very popular for real-time path planning because of its advantages in solving complex robot motion planning problems [13]. This algorithm is decomposed of two stages, the learning stage, and the query stage.

- In the learning stage, the algorithm builds an undirected graph of nodes and edges between a set of randomly generated nodes of the environment with collision-free.
- The second stage, the query phase, the algorithm explores all possible connections from a given start node S to a goal node G based on the developed graph of stage one.

Several attributes affect the performance of PRM. For example, the number of possible nodes in the environment, the allowable distance between each node, and many others. The classic PRM algorithm is used to calculate the shortest path. Nevertheless, it cannot change the node locations to achieve other goals such as path safety or smoothness. An example of a PRM that has thirty random points is shown in Figure 3. One of the main disadvantageous of PRM is that if the created random points are not fairly distributed on the environment as shown in Figure 3 there is no guarantee that the path to be found shall be an optimal one. The creation of a dense road map might be a solution but it is going to be computationally expensive. Another way to enhance the generated road map is to control the connection distance.

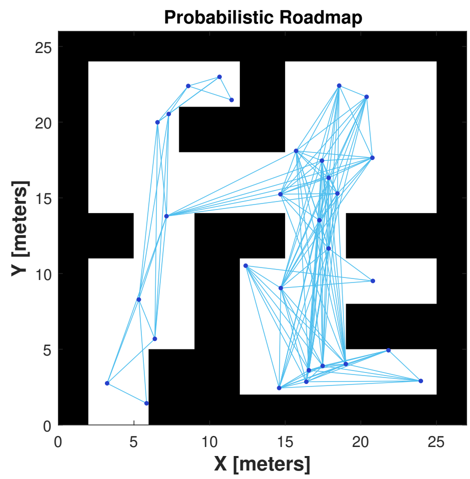

Figure 3: An example of a PRM.

3. Path Planning

The path planning problem is a common problem that can be found in logistics transportations, car navigation systems, computer communication networks, and personal or public evacuating decision systems through disasters [14]. Finding an optimal or near-optimal path is very important either for graph theory applications or network design [15]. In simple, optimal path algorithms are divided into two main categories: the optimum algorithm and the heuristic algorithm [16]. The optimum algorithm can find an optimal path. However, the time complexity will increase when having a complex PRM and not efficient for real-time navigation systems [17]. While heuristic and metaheuristic, algorithms can find an optimal or near-optimal path and can explore the search space efficiently compared to the optimum algorithm [18]. There are many heuristic algorithms such as A^* [19], the local multiresolution search algorithm [20], and the best-first search algorithm [21]. Heuristic algorithms can reduce the difficulties of searching for an optimal path. Meta-heuristic algorithms such as simulating annealing [22], Tabu Search [23] , and great deluge [24] show an excellent performance while searching for an optimal path. Few researchers have investigated their algorithms on complex maps [25]. In this paper, we investigated the performance of the LAHC algorithm over simple and complex PRM, and compare its performance with a well-known heuristic search algorithm, which is A^*.

3.1. The A^* Algorithm

A^* is a well-known informed search algorithm (ISA), or sometimes we call it best-first search (BFS). A^* finds the optimal path between a source point to a destination point using a weighted graph (i.e., Tree). Thus, starting from a certain node S of a graph to a node G, A^* explores the graph to find a path having the shortest distance traveled. The A^* search algorithm was developed as part of the Shakey project, which aims at the construction of a mobile robot system that is autonomous. A^* uses the Graph Traverser algorithm [26] for Shakey's path planning [27]. Although the Graph Traverser is only directed by a heuristic function $h(n)$. This technique ignores the estimated distance from node n, the distance from the start node to the intermediate node n. In [28], author suggested that both the distance between S to n and the distance between n and G need to be included, $g(n)+h(n)$. A^* maintains a tree of all possible paths created from S and spreading these paths one edge at a time until its performance criterion is fulfilled. A^* determines which path to extend based on some criteria (i.e., shortest distance). Thus, it checks all possible paths from a neighborhood node n to the goal node G. This process happens iteratively until the shortest path is found. Specifically, A* selects the paths that minimize Equation 1.

$$f(n) = g(n) + h(n) \tag{1}$$

where n is the next node on the path, $g(n)$ is the cost of the path from node S to n, and $h(n)$ is a heuristic function that estimates the cost of the cheapest path from n to G. Some of the well-known heuristic functions are the Manhattan distance, Euclidean distance, and Chebyshev distance.

The A^* search algorithm was successfully implemented and tested for many path planning applications [29–33]. It was reported in [34], that A^* is not acceptable concerning computation time especially for large maps. The author shows that the results are unpromising for the application of a robot path planning with about 60,000 cells. Therefore, further research was implemented to either enhance the A^* performance or adopt metaheuristics search algorithms [35] such as GAs [36–40], ant colony optimization [41], artificial bee colony [42] and PSO [43–45] algorithms.

3.2. The Late Acceptance Hill Climbing

In 2016, Burke and Bykov [47] proposed a simple, easy to implement, and effective local search algorithm called Late Acceptance Hill-Climbing (LAHC). It can be considered as an enhanced version of the Hill Climbing (HC) algorithm where the difference between HC and LAHC is the acceptance criterion that compares the new solutions to a solution obtained from the previous iterations. The basic idea of LAHC is to accept a new candidate solution if there is no improvement for a set of iterations. The main idea of LAHC comes from a late acceptance heuristic concept, which depends on the previously visited points in the search space. The algorithms work in the same manner as other local search algorithms such as Simulating Annealing (SA), Tabu Search (TS), and Great Deluge (GD). In simple, the LAHC starts with a random initial solution and either accepts or rejects the new solutions until the stop condition is met.

LAHC has an internal list of a fixed length called history length (L_h), which has the previous fitness values. While executing the LAHC algorithm, the variable $Iter_{idle}$ represents the status of the LAHC algorithm, which is increased by one if there is no improvement and back to zero if there is an improvement. The current solution is compared with the last fitness value in the L_h. LAHC will determine a virtual beginning v to determine the starting position to compare solutions inside L_h list. If the current solution is better than the v position in L_h, then the new solution is accepted and added to the first position in the list and remove the last solution in the list from the end of the list. The listed size is a single algorithmic value determined by the user. The size of L_h will enhance the overall performance of LAHC [47]. The pseudo-code of LAHC is shown in Algorithm 2.

```
Input: start, goal(n), h(n), expand(n)
Output: path
if goal(start) = true then
  | return makePath(start)
end
open ← start
closed ← ∅
while open ≠ ∅ do
  | sort(open)
  | n ← open.pop()
  | kids ← expand(n)
  | forall kid ∈ kids do
  |   | kid.f ← (n.g + 1) + h(kid)
  |   | if goal(kid) = true then return makePath(kid)
  |   | if kid ∩ closed = ∅ then open ← kid
  | end
  | closed ← n
end
return path
```

Algorithm 1: The A* Algorithm [46]

4. Experimental Results

This section explores and validates the obtained results of LAHC and A-star algorithms over two different PRM. Both algorithms have been implemented using MATLAB 2020. The fitness function for both algorithms is the accumulative distance between all nodes in a path. For our comparison, we used the same hardware infrastructure for all experiments.

Table 1 explores the parameters setting used in this paper. In both maps, we used 200 nodes that are located in feasible locations. The robot radius size is 0.2 meters. In this paper, we employed three different history lengths (i.e., 1, 100, and 500) for the LAHC algorithm.

The obtained results for simple PRM is shown in Table 2. The LAHC algorithm outperforms the A^* algorithm. LAHC reduces the distance obtained by A* by 5.23%. Moreover, we noticed that the size of the history length gives LAHC the ability to explore more search space. However, the execution time for A* is less than LAHC. Figures 4 and 5 show the obtained path for A* and LAHC algorithms, respectively.

Input: start point, goal point
Output: path
Generate initial path p
Determine History Length L_h
for k=0 *to* $L_h - 1$ **do**
| Calculate $f_k = fitness(p)$
end
while *(Iter < MaxIter) and* (I_{idle} > *Iter×0.2)* **do**
Generate a new candidate path P^*
Calculate the fitness for P^*
if *fitness*(p^*) $\geq$ *fitness* (*p*) **then**
| $Iter_{idle}$=$Iter_{idle}$+1
end
else
| $Iter_{idle}$= 0
end
calculate the virtual beginning $v := Iter \mod L_h$
if *fitness*(p^*) $< f_v$ *or fitness* (p^*) $\leq$ *fitness* (*P*) **then**
| Accept the candidate solution, p=p^*
end
else
| Reject the candidate solution, p=p
end
if *fitness*(p)<f_v **then**
| Then update the fitness array $f_v := fitness$(p)
end
Iter=Iter+1;
end
return path

Algorithm 2: The Late Acceptance Hill Climbing [48]

Table 1: Parameters setting.

	Attribute	Value(s)
General setting	Number of nodes	200
	Robot Radius	0.2
LAHC	Max iteration	1000
	History length (L_h)	1, 100, 500

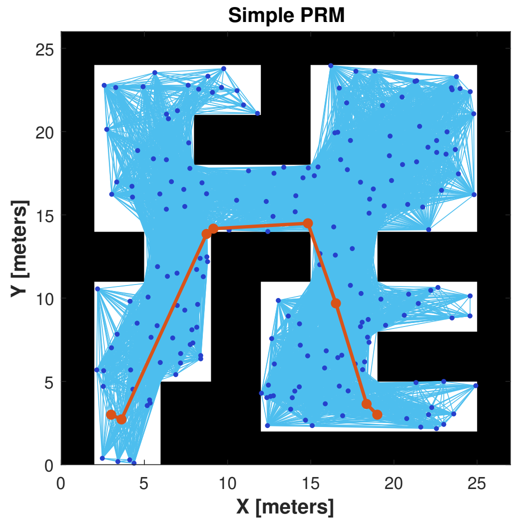

Figure 4: Best path obtained by A* for simple PRM.

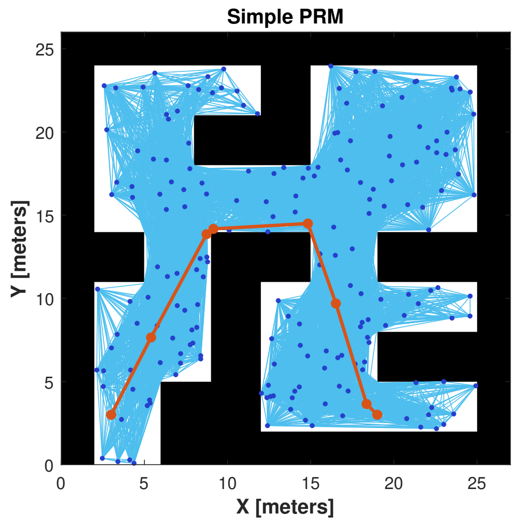

Figure 5: Best path obtained by LAHC for simple PRM.

Figure 6 explores the performance of LAHC using different sizes of history length. It is obvious increasing the size of history length enhanced the overall performance of LAHC. However, based on the results reported in Table 2, the execution time depends on history length size. Table 3 shows the obtained results for complex PRM. The performance of A^* is better than LAHC with history length equals 1, while the performance LAHC outperforms A^* with the larger history length size. Figures 7 and 8 simulate the best-obtained paths for A^* and LAHC algorithms, respectively.

Figure 9 explores the performance of LAHC with different history length sizes. The larger history length needs more execution time. Moreover, the performance of LAHC with history length equals 500 can efficiently explore the search space. Table 4 shows a statistical analysis based on Wilcoxon statistical test. A threshold value equals to 0.05. In this table, if P-value is less than 0.05 means there is a statistical difference between the obtained results. The performance of LAHC depends on the L_h size, where all obtained p-values are less than 0.05.

5. Conclusions and Future Work

In this paper, we investigated the performance of LAHC and A^* for indoor unmanned aerial vehicle (UAV) path planning problems. The complexity of this problem increases if the search space (i.e., PRM) is complex with many obstacles. We compared the performance of LAHC and A^* over simple and complex PRMs. The performance of LAHC outperforms A^* in both cases. Moreover, we examine the performance of LAHC with different sizes of history length (L_h) array (i.e., 1, 100, and 500). The obtained results show that the higher L_h

Table 2: Results for simple PRM.

	History length	Fitness value	Time (sec)
A*	—	38.78	0.14
LAHA	1	38.65	1.10
	100	37.81	3.62
	500	36.75	6.79

Table 3: Results for complex PRM.

	History length	Fitness value	Time (sec)
A*	—	91.432	0.5726
LAHA	1	102.350	2.0341
	100	90.591	4.1526
	500	89.671	7.2283

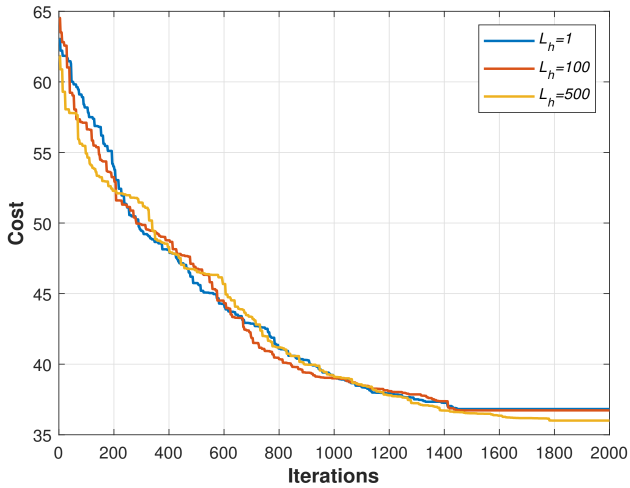

Figure 6: LAHC convergence with different history length for simple map.

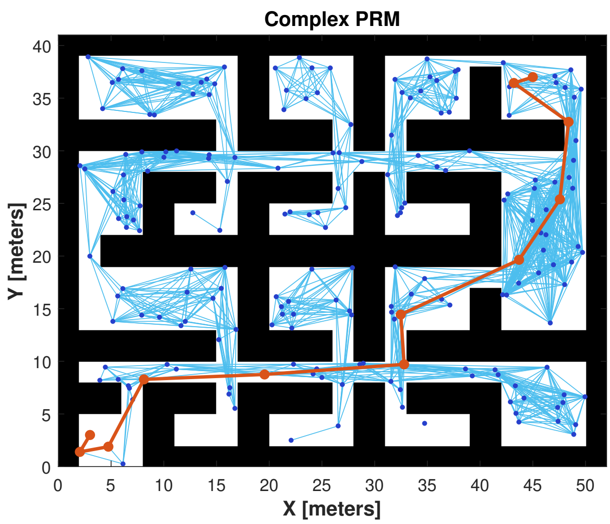

Figure 7: Best path obtained by A* for complex PRM.

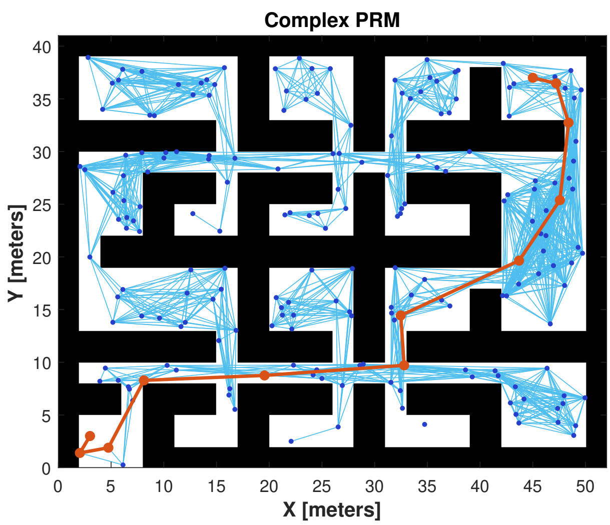

Figure 8: Best path obtained by LAHC for complex PRM.

size the better performance of LAHC. In the future, we plan to investigate a more complex environment and enhance the

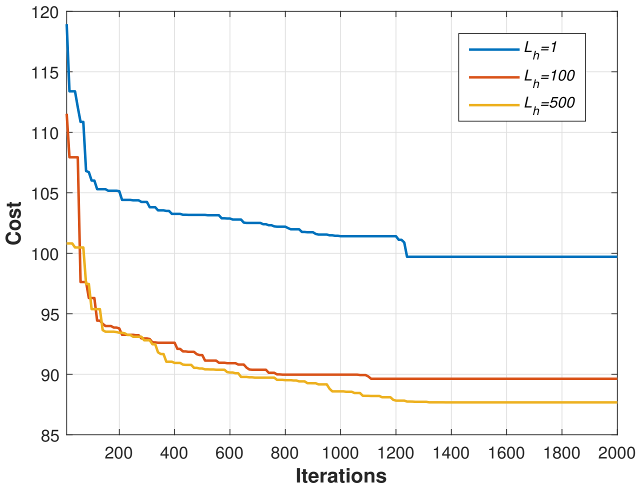

Figure 9: LAHC convergence with different history length for complex map.

Table 4: Statistical analysis between different L_h sizes.

	P-value
L_h=1 vs. L_h=100	0.027
L_h=1 vs. L_h=500	0.004
L_h=100 vs. L_h=500	0.013

performance of LAHC by tuning the L_h size in an automated way based on the status of the current performance of LAHC.

Acknowledgment

The authors would like to acknowledge the financial support for the Research Experiences for Undergraduates (REU) funded by the National Science Foundation (NSF) under grant number 1757940.

References

[1] T. Goetzendorf-Grabowski and M. Rodzewicz, “Design of uav for photogrammetric mission in antarctic area,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 231, no. 9, pp. 1660–1675, 2017.

[2] J. Kim, S. Kim, C. Ju, and H. I. Son, “Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications,” *IEEE Access*, vol. 7, pp. 105 100–105 115, 2019.

[3] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, “A compilation of uav applications for precision agriculture,” *Computer Networks*, vol. 172, p. 107148, 2020.

[4] H. T. Berie and I. Burud, “Application of unmanned aerial vehicles in earth resources monitoring: focus on evaluating potentials for forest monitoring in ethiopia,” *European Journal of Remote Sensing*, vol. 51, no. 1, pp. 326–335, 2018.

[5] R. Koslowski and M. Schulzke, “Drones along borders: Border security uavs in the united states and the european union,” *International Studies Perspectives*, vol. 19, July 2017.

[6] G. Sylvester, G. Rambaldi, D. Guerin, A. Wisniewski, N. Khan, J. Veale, and M. Xiao, “E-agriculture in action: drones for agriculture. food and agriculture organization of the united nations and international telecommunication union, bangkok, thailand,” *Bangkok: Food and Agriculture Organization of the United Nations and International Telecommunication Union. Retrieved July*, vol. 19, p. 2018, 2018.

[7] Ruijie He, S. Prentice, and N. Roy, “Planning in information space for a quadrotor helicopter in a gps-denied environment,” in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 1814–1820.

[8] C. Huang, Y. Lan, Y. Liu, W. Zhou, H. Pei, L. Yang, Y. Cheng, Y. Hao, and Y. Peng, “A new dynamic path

planning approach for unmanned aerial vehicles," *Complexity*, vol. 2018, pp. 1–17, November 2018.

[9] R. M. Santiago, A. L. De Ocampo, A. Ubando, A. Bandala, and E. Dadios, "Path planning for mobile robots using genetic algorithm and probabilistic roadmap," December 2017, pp. 1–5.

[10] Y. Chen, F. Su, and L.-C. Shen, "Improved ant colony algorithm based on prm for uav route planning," vol. 21, pp. 1658–1666, March 2009.

[11] D. Zhao and J. Yi, "Robot planning with artificial potential field guided ant colony optimization algorithm," in *Advances in Natural Computation*, L. Jiao, L. Wang, X. Gao, J. Liu, and F. Wu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 222–231.

[12] Y. Wu, "An obstacle based probabilistic roadmap method for path planning," 1996.

[13] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, pp. 566–580, September 1996.

[14] M. Xue, "Optimal path selection of vehicle route scheduling under bad weather factors," *Computer Simulation*, vol. 32, no. 2, pp. 210–212, 2015.

[15] R. Sotnezov, "Genetic algorithms for problems of logical data analysis in discrete optimization and image recognition," *Pattern Recognition and Image Analysis*, vol. 19, no. 3, pp. 469–477, 2009.

[16] B. Yan, T. Chen, X. Zhu, Y. Yue, B. Xu, and K. Shi, "A comprehensive survey and analysis on path planning algorithms and heuristic functions," in *Science and Information Conference*. Springer, 2020, pp. 581–598.

[17] W. Qiuping, S. Rong, and Z. Qi, "Optimal path selection of slow traffic based on gis network analysis," *Journal of Xi'an University of Architecture & Technology*, vol. 45, no. 5, pp. 668–674, 2013.

[18] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.

[19] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014.

[20] S. Behnke, "Local multiresolution path planning," in *Robot Soccer World Cup*. Springer, 2003, pp. 332–343.

[21] C. M. Dellin and S. S. Srinivasa, "A unifying formalism for shortest path problems with expensive edge evaluations via lazy best-first search over paths with edge selectors," in *Twenty-Sixth International Conference on Automated Planning and Scheduling*, 2016.

[22] Q. Zhu, Y. Yan, and Z. Xing, "Robot path planning based on artificial potential field approach with simulated annealing," in *Sixth International Conference on Intelligent Systems Design and Applications*, vol. 2. IEEE, 2006, pp. 622–627.

[23] W. Khaksar, T. S. Hong, M. Khaksar, and O. R. E. Motlagh, "Sampling-based tabu search approach for online path planning," *Advanced Robotics*, vol. 26, no. 8-9, pp. 1013–1034, 2012.

[24] E. Akar, H. R. Topcuoglu, and M. Ermis, "Hyper-heuristics for online uav path planning under imperfect information," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2014, pp. 741–752.

[25] U. F. Siddiqi, Y. Shiraishi, and S. M. Sait, "Multi-objective optimal path selection in electric vehicles," *Artificial Life and Robotics*, vol. 17, no. 1, pp. 113–122, 2012.

[26] J. E. Doran and D. Michie, "Experiments with the graph traverser program," in *Proceedings of the Royal Society of London, Series A. Mathematical and Physical Sciences*, vol. 294, no. 1437, September 1966.

[27] N. J. Nilsson, *The Quest for Artificial Intelligence*. Cambridge University Press, 2009.

[28] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *SIGART Bull.*, no. 37, pp. 28–29, Dec. 1972.

[29] M. Korkmaz and A. Durdu, "Comparison of optimal path planning algorithms," in *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)*, February 2018, pp. 255–258.

[30] V. Kunchev, L. Jain, V. Ivancevic, and A. Finn, "Path planning and obstacle avoidance for autonomous mobile robots: A review," in *Knowledge-Based Intelligent Information and Engineering Systems*, B. Gabrys, R. J. Howlett, and L. C. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 537–544.

[31] O. Hachour, "Path planning of autonomous mobile robot," *International journal of systems applications, engineering & development*, vol. 2, no. 4, pp. 178–190, 2008.

[32] M. Radovnikovich, K. C. Cheok, and P. Vempaty, "Comparison of optimal path planning algorithms for an autonomous mobile robot," in *2011 IEEE Conference on Technologies for Practical Robot Applications*, April 2011, pp. 35–39.

[33] P. Raja and S. Pugazhenthi, "Optimal path planning of mobile robots: A review," *International Journal of Physical Sciences*, vol. 7, no. 9, pp. 1314 – 1320, 2012.

[34] F. Duchoň, D. Huňady, M. Dekan, and A. Babinec, "Optimal navigation for mobile robot in known environment," in *Robotics in Theory and Practice*, ser. Applied Mechanics and Materials, vol. 282. Trans Tech Publications Ltd, March 2013, pp. 33–38.

[35] A. Hussein, H. Mostafa, M. Badrel-din, O. Sultan, and A. Khamis, "Metaheuristic optimization approach to mobile robot path planning," in *2012 International Conference on Engineering and Technology (ICET)*, October 2012, pp. 1–6.

[36] A. Ismail, A. Sheta, and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment," *Journal of Computer Science*, vol. 4, no. 4, pp. 341–344, 2008.

[37] S. C. Yun, V. Ganapathy, and L. O. Chong, "Improved genetic algorithms based optimum path planning for mobile robot," in *2010 11th International Conference on Control Automation Robotics Vision*, Dec 2010, pp. 1565–1570.

[38] R. M. C. Santiago, A. L. D. Ocampo, A. T. Ubando, A. A. Bandala, and E. P. Dadios, "Path planning for mobile robots using genetic algorithm and probabilistic roadmap," in *2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Dec 2017, pp. 1–5.

[39] M. Wu, E. Chen, Q. Shi, L. Zhou, Z. Chen, and M. Li, "Path planning of mobile robot based on improved genetic algorithm," in *2017 Chinese Automation Congress (CAC)*, Oct 2017, pp. 6696–6700.

[40] Y. Liang, F. Hong, Q. Lin, S. Bi, and L. Feng, "Optimization of robot path planning parameters based on genetic algorithm," in *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, July 2017, pp. 529–534.

[41] J. Chen, F. Ye, and T. Jiang, "Path planning under obstacle-avoidance constraints based on ant colony optimization algorithm," in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, October 2017, pp. 1434–1438.

[42] N. ALPKIRAY, Y. TORUN, and O. KAYNAR, "Probabilistic roadmap and artificial bee colony algorithm cooperation for path planning," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Sep. 2018, pp. 1–6.

[43] D.-W. Gong, J.-H. Zhang, and Y. Zhang, "Multi-objective particle swarm optimization for robot path planning in environment with danger sources," *JCP*, vol. 6, pp. 1554–1561, August 2011.

[44] R. Grandi, "Coordination and controm of autonomous mobile robot swarms by using particle swarm optimization algorithm and consensus theory," Master's thesis, Alma Mater Studiorum - Universita di Bologna, May 2013.

[45] M. S. Alam, M. U. Rafique, and M. U. Khan, "Mobile robot path planning in static environments using particle swarm optimization," *International journal of computer science and electronics engineering (IJCSEE)*, vol. 3, pp. 253–257, 2015.

[46] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[47] E. K. Burke and Y. Bykov, "The late acceptance hill-climbing heuristic," *European Journal of Operational Research*, vol. 258, no. 1, pp. 70 – 78, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221716305495

[48] A. Goerler, F. Schulte, and S. Voß, "An application of late acceptance hill-climbing to the traveling purchaser problem," in *Computational Logistics*, D. Pacino, S. Voß, and R. M. Jensen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 173–183.