



A Modified Particle Swarm Optimization for Autonomous UAV Path Planning in 3D Environment

Golam Moktader Nayeem^{1(✉)}, Mingyu Fan^{1(✉)}, Shanjun Li²,
and Khalil Ahammad³

¹ School of Computer Science and Engineering,
University of Electronic Science and Technology of China,
Chengdu 611731, People's Republic of China
gmnayeem@std.uestc.edu.cn, ff98@163.com

² Key Laboratory of Information Network Security, Ministry of Public Security,
Shanghai 201804, People's Republic of China
shanjunli@yahoo.com

³ Department of Computer Science and Engineering, Comilla University,
Cumilla, Bangladesh
khalil.cou.cse@gmail.com

Abstract. Path planning is an important aspect of an autonomous Unmanned Aerial Vehicle (UAV). As finding the best path is a non-deterministic problem, meta-heuristic algorithms proved to be a better choice in recent years. Particle Swarm Optimization (PSO) is one of the widely applied meta-heuristic algorithms for non-deterministic problem due to simplicity and ease of implementation. However, the lack of diversity in the particles in PSO algorithm generates a low-quality path for UAV. In this paper, we presented a modified PSO algorithm called n-PSO. In the algorithm, a dynamic neighborhood approach is proposed to improve the diversity of the particles. The n-PSO algorithm is applied to UAV path planning and simulated in a 3D environment. We compared the algorithm with two widely used versions of PSO for UAV path planning. The proposed algorithm showed significant improvement in particles diversity that plays an important role to produce better UAV path. At the end, we presented a time cost analysis of the algorithm for UAV path planning.

Keywords: PSO · UAV · Path planning · 3D environment

1 Introduction

The development of Autonomous Unmanned aerial vehicles (UAV) is of high interest to many researchers in recent years due to its diverse application in both military and civilian environment [1]. Some widely used applications are surveillance, search and rescue, disaster management, target search, goods delivery, wildlife research, etc. [2]. Path planning is an indispensable part of an

autonomous UAV system and aims to find the optimum path from start point to end point while satisfying the required constraints such as distance, time, power consumption, safety, etc. Path planning algorithm can be classified as offline and online. In offline path planning, an optimized path is planned before UAV fly for the known destination. On the other hand, online path planning aims to generate a dynamic path on the basis of offline reference path and real-time data of the track [3]. In this work, we considered offline path planning. In conventional path planning problem, one of the basic parameter used to calculate the best path is using the minimum distance traveled like as Traveling Salesman Problem (TSP). But constraints such as obstacle, average altitude, maximum turning angle, etc. is considered in UAV path planning. Therefore the problem becomes complex and characterized as a multi-objective optimization problem [4].

Meta-heuristic methods have been evolved to solve multi-objective optimization problem more efficiently than other available methods such as classical method, heuristic method, etc. [5]. PSO is a nature-inspired population-based meta-heuristic approach that has gained much attention in solving multi-objective optimization problems [6]. PSO deals with a set of solutions inspired by a group of an agent called particle [7]. The major advantages of using PSO are: (a) computationally inexpensive; (b) simple and easy implementation; (c) minimum interaction between particles as it uses its local best and global best information to calculate the new position; and (d) distributed processing as no mutation operation is required to generate the next generation of solution [8]. However, one of the major issues of PSO is the lack of diversity in the particle. Lack of diversity in the particles causes premature convergence and fall for local minima. As a result, a low quality solution is generated for UAV path planning [9]. In this paper, we focus on this particular problem of PSO in UAV path planning. We presented a modified PSO that produces a better solution for UAV path planning by increasing particles diversity. The main contributions of this work are: (a) We introduced a modified version of PSO named n-PSO; (b) We applied the n-PSO algorithm for UAV path planning with static obstacles in a 3D environment; and (c) We compared the n-PSO with other standard versions of PSO used for UAV path planning.

In the subsequent sections, we first give an overview of existing algorithms. In Sect. 3, we present a detailed mathematical model of the problem. An overview of the proposed model is described in Sect. 4. In Sect. 5, we present the experimental outcome of the proposed model. Lastly, we conclude the paper in Sect. 6.

2 Related Work

In recent years, a number of offline UAV path planning algorithms have been proposed based on PSO. Garcia *et al.* proposed a distributed PSO algorithm for exploration purpose for a search and rescue mission [10]. In the proposed model, PSO is used to generate the waypoints for the UAVs whereas lawnmower algorithm is used for the exploration task for UAVs. Also a new parameter is introduced instead of r_1 and r_2 to decrease the randomness of the algorithm in

order to speed up the searching task of multiple targets. However, the author did not consider the diversity factor of the particles in the algorithm and the algorithm is implemented in a 2D environment without any obstacle. Huang *et al.* proposed a modified PSO called GBPSO for UAV path planning [4]. A competition strategy is introduced for selecting the global best position that helps to increase the diversity in the particles and speed up the global search process. A variant of PSO named CPSO for 2D offline path planning is proposed in [11]. In the proposed model, chaotic maps such as Singer map is used to determine the values of two basic parameters of PSO i.e. exploration and exploitation which create more diversity among the particles. However, the author did not provide the effect of a chaotic map in the particles diversity. Roberge *et al.* proposed a mathematical model and a cost function for UAV in a 3D environment and developed a parallel implementation method for UAV path planning [12]. In [9], a modified PSO named vPSO is proposed that uses Gaussian probability density function for mutation strategy in Gaussian mutation based PSO algorithm [13]. The author showed that increased diversity in the particles produces a good quality solution for UAV path planning. In addition, a number of variation of standard PSO have been proposed so far to increase the diversity in the particle for general purpose. A new parameter named constriction factor is introduced to improve the stability of PSO by Clerc [14]. Shi *et al.* introduced a time-varying inertia weight to improve the particle diversity and to avoid premature convergence [15]. To improve the diversity in the particles Sugathan *et al.* added a dynamic neighborhood parameter in PSO [16]. However, the author did not explicitly define the parameter and the algorithm is simulated with a general cost function. From the above discussion, we conclude that the effect of particle diversity in generating a good quality solution for UAV path planning requires more research attention. In our work, we introduced a dynamic neighborhood parameter named $nhbest$ that increases the diversity in the particles and generates a better path for an autonomous UAV. We provided a comprehensive comparison with existing well-known versions of PSO used for UAV path planning. i.e. c-PSO [14], w-PSO [15].

3 Mathematical Model

3.1 Unmanned Ariel Vehicle (UAV)

A UAV can be classified as an autonomous robot that can move in a 3-Dimensional Space [17]. Let consider U represent UAV and S is a 3D terrain or search space where the UAV is going to move around at discrete time step t . The UAV at time state t can be expressed in 3-dimensional space and can be presented as $U_t = (x_t, y_t, z_t)$ where x_t , y_t represent axis position and z_t represent the altitude of the UAV at time t . However, the UAV flight dynamics is ignored in this work.

3.2 Environment

Defining the environment is an important part of UAV path planning. In this paper, we considered a 3D environment with multiple spherical static obstacles treated as a no-fly zone (NFZ). The environment is represented in a discrete form in the algorithm [18]. We consider a rectangular matrix $S \in R^3$ represented by $X \cdot Y \cdot Z$ where X and Y represent the x-axis and y-axis respectively and Z represent the altitude. Each cell is represented by $C = x_i, y_i, z_i$ where $x_i \in X$, $y_i \in Y$ and $z_i \in Z$. The NFZ is represented as follows:

$$NFZ = \begin{bmatrix} x_1 & y_1 & z_1 & d_1 \\ x_2 & y_2 & z_2 & d_2 \\ \dots & \dots & \dots & \dots \\ x_n & y_n & z_n & d_n \end{bmatrix} \quad (1)$$

where d_i represent the diameter of the i th object.

3.3 UAV Path

Path planning refers to generating the waypoints for UAVs. The waypoints are separated by equal time steps [12]. However, the discrete path generated by the algorithm is not follow-able by the UAV dynamics and kinematics. Therefore, the generated path needs to be smoothed before it is fed to the UAV flight controller. In this work, we used basis spline or bspline cubic interpolation formula with smoothing factor 0.2 to smooth the path generated by the algorithm [19]. The path is denoted by a matrix where each row represents the i th waypoint.

$$trajectory = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \dots & \dots & \dots \\ x_n & y_n & z_n \end{bmatrix} \quad (2)$$

3.4 Cost Function

Defining the appropriate cost function is an import part of UAV path planning. The optimization algorithm finds the best solution based on the cost function. Optimization problem like TSP uses path length to find the shortest path that traverses all the cities. In the case of UAV path planning, the problem becomes more complex due to multiple constraints such as shortest path, obstacle avoidance, minimum altitude, sharp turn, etc. The algorithm searches for a solution with minimum cost. In this work, we have considered two constraints for UAV path planning i.e. shortest path and obstacle avoidance. We define our cost function as follows:

$$F_{cost} = C_{Path} + C_{NFZ} \quad (3)$$

where C_{Path} calculates the generated path length and C_{NFZ} penalize if the path collide with the no-fly zone. The path length is calculated using the following

Euclidean distance formula:

$$C_{Path} = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (4)$$

where n is the number of waypoints generated, x_i, y_i, z_i are the axis position of UAV in 3D space. To evaluate whether the generated path collides with NFZ, we first calculate the distance from each waypoint from the NFZ center. If the distance is less than the diameter of the NFZ, then the function penalize heavily. The C_{NFZ} function is calculated using the following equations:

$$Collide = 1 - \sum_{i=1}^{n_{NFZ}} \frac{\sum_{j=1}^{n-1} \sqrt{(x_j - NFZ_i^x)^2 + (y_j - NFZ_i^y)^2 + (z_j - NFZ_i^z)^2}}{d_i} \quad (5)$$

$$C_{NFZ} = \text{argMax}(Collide, 0) \quad (6)$$

where NFZ_i^x, NFZ_i^y and NFZ_i^z are 3D center coordinate of i th NFZ, n_{NFZ} is total number of NFZ and d_i is the diameter of i th NFZ.

4 n-PSO for Path Planning

4.1 PSO Framework

PSO is a population-based evolutionary algorithm which was first introduced in [20]. The algorithm is inspired by animal common social behavior found in nature like as birds flock. The basic idea of PSO is as stated: “*people learn to make sense of the world by talking with other people about it*” [7]. The major advantage of PSO is simplicity and problem-solving ability. Since its introduction, PSO has been used in numerous applications for solving optimization problem. The idea of PSO is a number of candidate solutions called particles are taken into consideration and the position of each particle have been updated on the basis of particle best position (local best) and neighbors or swarms best (global best) position iteratively until all the particles converge to an optimal solution. More precisely we can describe that in PSO each particle has treated as a point in an N-dimensional space. The position of each particle is updated or adjusted on the basis of its flying experience and other particles flying experience. Each particle modifies its position on the basis of the information such as current position, current velocity, distance between particle best positions with the current position and distance between global best positions with the current position [14]. Let $P(t)$ is a population of n particles is a multidimensional space at time step t , where $P(t) = \{P_1(t), P_2(t), \dots, P_i(t), \dots, P_n(t)\}$. $P_i(t)$ is the i th particle in d -dimensional space at time step t , where $P_i(t) = \{X_{i,1}(t), X_{i,2}(t), \dots, X_{i,d}(t)\}$ is i th particle position in d -dimensional space. The velocity of i th particle at time step t can be represented as $V_i(t) = \{V_{i,1}(t), V_{i,2}(t), \dots, V_{i,d}(t)\}$. As outlined in [7], the equation for calculating the new position of i th particle $P_i(t)$ are as follows:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (7)$$

$$V_i(t+1) = \omega(t)V_i(t) + c_1r_1(P_l(t) - X_i(t)) + c_2r_2(P_g(t) - X_i(t)) \quad (8)$$

where the terms are as follows:

- $V_i(t)$: Particle velocity at time t .
- $\omega(t)$: Inertia at time t .
- c_1 and c_2 : Constants that used for personal influence and social influence respectively.
- r_1 and r_2 : Random values between 0 and 1.
- $P_l(t)$: Particle local best.
- $P_g(t)$: Particle global best.

The value of $\omega(t)$ is calculated using the following formula:

$$\omega(t) = \omega_{max} - t \cdot \frac{\omega_{max} - \omega_{min}}{MaxIter} \quad (9)$$

where ω_{max} , ω_{min} and $MaxIter$ are final inertia, initial inertia and maximum number of iterations respectively. The particle local best position and global best position vector is computed using the following equations:

$$P_l(t) = \begin{cases} P_l(t-1) & \text{if } F_{cost}(X_i(t)) \geq F_{cost}P_l(t-1) \\ X_i(t) & \text{if } F_{cost}(X_i(t)) \leq F_{cost}(P_l(t-1)) \end{cases} \quad (10)$$

$$P_g(t) = argmin_{P_l(t)} F_{cost}(P_l(t)) \quad (11)$$

4.2 Improved PSO

The two most well-known improvements of PSO are constriction factor PSO (c-PSO) [14] and time-varying inertia weight PSO (w-PSO) [15]. Both these methods are proposed to balance between two important characteristics of PSO algorithm i.e. exploitation and exploration. However, PSO still lacks particles diversity. Diversity in the particles is an important part of PSO in order to produce a good quality solution. In the proposed model called n-PSO, we introduced a new parameter based on the neighborhood approach proposed in [16]. The basic idea of our proposed algorithm is, instead to directly jump into the global best position; the particle creates a dynamic neighborhood and uses the neighborhood best position for exploration. The radius of the neighborhood is initially set to 1 and will increase linearly until all particles become a single neighborhood. As a result, the particles converge to the solution in a distributed manner and able to avoid local minima. The time-varying size of the neighborhood is calculated by the following equations:

$$nhsize = nhsize + \frac{N_{pop}}{MaxIt} \quad (12)$$

$$nhbest_i = min_{P_{nh_i}} F_{cost}(P_{nh_i}) \quad (13)$$

where $nhsize$, N_{pop} and $MaxIt$ are neighborhood size, total number of population and maximum number of iterations. The neighborhood best position can be calculated by equation no. (13) where P_{nh_i} is the particle position of i th neighborhood. The proposed algorithm is presented as follows:

Algorithm 1. n-PSO Algorithm

```

1: Create model with  $n$  static particles
2: Evaluate the cost based on initial particles position
3: Calculate the particle local best position  $P_l(t)$ 
4: Set  $nhsz$  = 1 and update the  $nhbest_i$  from  $P_l(t)$ 
5: Set the parameter  $\omega_{max}$ ,  $\omega_{min}$ ,  $c_1$  and  $c_2$ 
6: for  $It = 1$  to  $MaxIter$  do,
7:   for  $i = 1$  to  $N_{pop}$  do
8:     Calculate  $\omega(t)$ 
9:     Calculate particle velocity  $V_i(t)$ 
10:    Calculate particle new position  $X_i(t + 1)$ 
11:    Calculate the cost of particle new position  $F_{cost}$ 
12:    Update particle local best position  $P_l(t)$ 
13:    Update global best position  $P_g(t)$ 
14:    for  $i = 1$  to  $\frac{N_{pop}}{nhsz}$  do
15:      for  $nhp = 1$  to  $nhsz$  do
16:        Evaluate  $nhbest_i$ 
17:        Select the  $nhbest_i$ 
18:    Update  $nhsz$ 
19: Output the best path

```

5 Experiment and Discussion

We implemented our algorithm using Python 3.7 in a PC (CPU Core i5 3.3 GHz & RAM 8 GB). In the c-PSO algorithm, we have set the value of constriction factor (k) to 0.729 to generate better result [9]. In the w-PSO algorithm, the range of inertia weight (ω) is set from 0.6 to 0.2 [15]. In n-PSO, we kept the similar inertia parameter to incorporate the advantages of w-PSO. The value of c_1 and c_2 were set to 2.05 for c-PSO, 2.0 and 1.5 respectively in w-PSO and n-PSO. We first implemented the algorithm in a 2D environment. Figure 1, shows a 2D UAV path generated by the proposed n-PSO algorithm.

Table 1. Parameter settings for test cases.

Test case	Search space dimension	No. of NFZ	No. of populations	No. of iterations
1	20 × 20	9	100	100
2	20 × 20	9	200	150
3	40 × 40	15	100	100
4	40 × 40	15	200	150
5	60 × 60	20	100	100
6	60 × 60	20	200	150
7	80 × 80	30	100	100
8	80 × 80	30	200	150
9	100 × 100	40	100	100
10	100 × 100	40	200	150

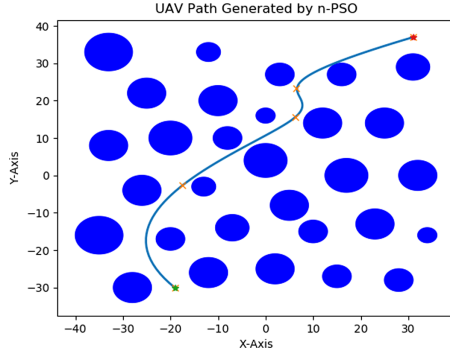


Fig. 1. A 2D UAV path generated by n-PSO algorithm

In order to compare the proposed algorithm with the other two algorithms, we have set 10 test cases for the simulation. We run each test cases 5 times for each of the algorithms and calculated the average best cost and diversity. The parameters for the test cases are given in Table 1. The performance of the proposed algorithm is presented in Table 2. In every case, n-PSO have generated a better solution than other two algorithms for UAV path planning. The improvement become more significant for larger search space size, number of iterations and number of populations.

Table 2. Comparison of different algorithm with respect to Best Cost.

Test case	c-PSO	w-PSO	n-PSO
1	20.565	20.507	20.451
2	20.465	20.557	20.453
3	67.256	49.169	45.515
4	59.571	42.031	39.966
5	90.811	63.839	62.929
6	68.895	64.940	57.845
7	114.045	94.864	94.494
8	103.443	93.145	91.761
9	140.929	138.029	136.131
10	135.557	134.456	120.599

In Fig. 2, a comparison is presented to show the convergence rate of the algorithms. From the figure, we can see that n-PSO converges slowly compared to both c-PSO and w-PSO. But, eventually it has converged to more accurate solution for an equal number of iterations. In order to measure the particles

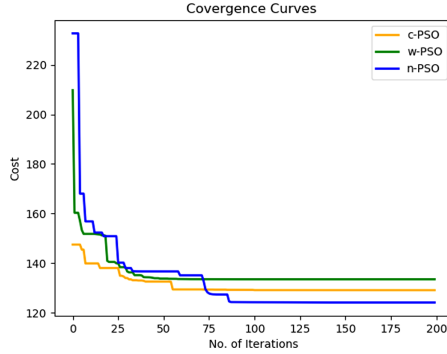


Fig. 2. Comparisons of n-PSO with c-PSO and w-PSO with respect to the convergence rate

diversity of the proposed algorithm, we used the following formula proposed by Olorunda [21]:

$$Diversity\ D(t) = \frac{1}{N_{pop}} \sum_{i=1}^{N_{pop}} \sqrt{\sum_{j=1}^{Dim} (x_{ij}(t) - \overline{x_j(t)})^2} \tag{14}$$

where N_{pop} is the total population size, $x_{ij}(t)$ is i th particle in j dimension and $\overline{x_j(t)}$ is average of j th dimension. The value of $\overline{x_j(t)}$ is can be calculated by following equation:

$$\overline{x_j(t)} = \frac{1}{N_{pop}} \sum_i^{N_{pop}} x_{ij} \tag{15}$$

In Table 3, we presented a comparison of particle diversity between n-PSO with others. It shows that the particles diversity of the proposed algorithm has increased significantly. The dynamic neighborhood approach prevent the algorithm to converge quickly to the global optima and increases the particles diversity. Figure 3 shows a comparison of particles diversity variation with respect to a number of iterations between three algorithms. It shows that the diversity in the particles has increased significantly in n-PSO.

In Fig. 4, we presented a time cost comparison of our proposed algorithm with the other two methods. From the figure, we can conclude that the new parameter did not put any significant computational load to the algorithm. Finally, we implement the n-PSO algorithm in a 3D environment with static obstacles. Figure 5 shows a UAV path generated by n-PSO in a 3D environment.

Table 3. Comparison of different algorithm with respect to Best Cost.

Test case	c-PSO	w-PSO	n-PSO
1	1.638	0.567	2.981
2	1.723	0.259	2.262
3	2.978	2.291	9.524
4	4.015	2.776	8.928
5	3.458	3.647	13.501
6	8.625	2.023	11.453
7	7.355	5.165	16.718
8	9.834	6.726	13.071
9	14.304	3.828	20.956
10	12.035	3.355	17.033

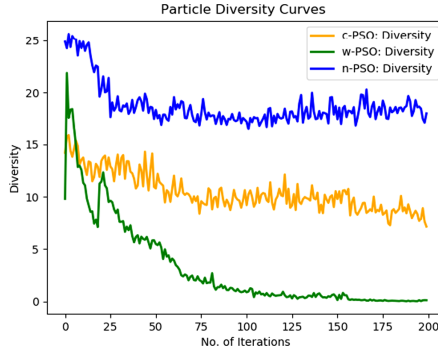
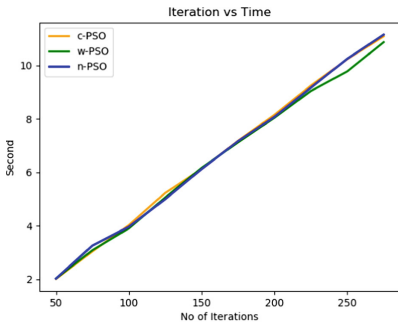
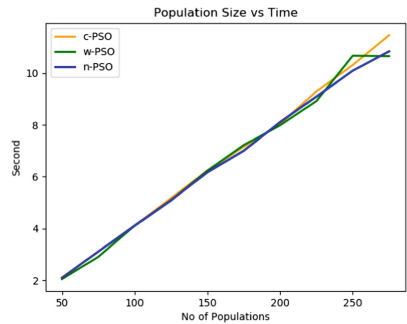


Fig. 3. Diversity variation versus no. of iterations



(a) Iteration versus Time.



(b) Population size versus Time

Fig. 4. Time cost analysis of n-PSO

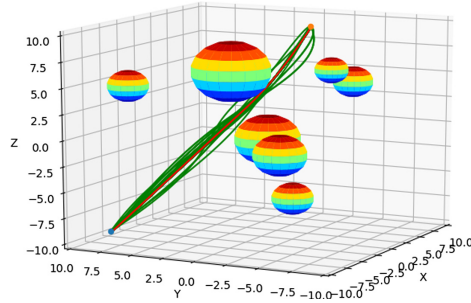


Fig. 5. An UAV path generated by n-PSO in 3D environment.

6 Conclusion

This paper presents a modified PSO based path planning solution for autonomous UAV in a 3D environment. A dynamic neighborhood-based approach is proposed to increase particle diversity. The improvement of the algorithm is verified with two well-known versions of PSO used for UAV path planning. The result showed that the particle diversity in n-PSO has increased significantly which enable n-PSO to successfully produce better UAV path than c-PSO and w-PSO. Also, the time cost analysis showed that the new parameter in the algorithm has less effect on the total execution time. However, we implemented and tested the algorithm in a small scale environment. Also, more rigorous analysis is required to prove that increased diversity in PSO will guarantee better UAV path which is our future area of interest. In addition to that, real-time path planning in 3D environment is also a key research area of UAV path planning.

Acknowledgment. This work is supported by Key Lab of Information Network Security, Ministry of Public Security, Shanghai 201804, China.

References

1. Zhao, Y., Zheng, Z., Liu, Y.: Survey on computational-intelligence-based UAV path planning. *Knowl.-Based Syst.* **158**, 54–64 (2018)
2. Sebbane, Y.B.: *Intelligent Autonomy of UAVs: Advanced Missions and Future Use*. Chapman and Hall/CRC, London (2018)
3. Cheng, Z., Wang, E., Tang, Y., Wang, Y.: Real-time path planning strategy for UAV based on improved particle swarm optimization. *JCP* **9**(1), 209–214 (2014)
4. Huang, C., Fei, J.: UAV path planning based on particle swarm optimization with global best path competition. *Int. J. Pattern Recogn. Artif. Intell.* **32**(06), 1859008 (2018)
5. Boussaid, I., Lepagnot, J., Siarry, P.: A survey on optimization metaheuristics. *Inf. Sci.* **237**, 82–117 (2013)
6. Zhang, Y., Wang, S., Ji, G.: A comprehensive survey on particle swarm optimization algorithm and its applications. *Math. Probl. Eng.* **2015**, 1–38 (2015)

7. Kennedy, J.: Swarm intelligence. In: Zomaya, A.Y. (ed.) *Handbook of Nature-Inspired and Innovative Computing*, pp. 187–219. Springer, Boston (2006). https://doi.org/10.1007/0-387-27705-6_6
8. Zafar, M.N., Mohanta, J.: Methodology for path planning and optimization of mobile robots: a review. *Procedia Comput. Sci.* **133**, 141–152 (2018)
9. Pehlivanoglu, Y.V.: A new particle swarm optimization method for the path planning of UAV in 3D environment. *J. Aeronaut. Space Technol.* **5**(4), 1–14 (2012)
10. Sanchez-Garcia, J., Reina, D., Toral, S.: A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario. *Future Gener. Comput. Syst.* **90**, 129–148 (2019)
11. Tharwat, A., Elhoseny, M., Hassanien, A.E., Gabel, T., Kumar, A.: Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm. *Cluster Comput.* **22**(2), 4745–4766 (2018). <https://doi.org/10.1007/s10586-018-2360-3>
12. Roberge, V., Tarbouchi, M., Labonté, G.: Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Trans. Ind. Inform.* **9**(1), 132–141 (2013)
13. Higashi, N., Iba, H.: Particle swarm optimization with Gaussian mutation. In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS 2003)* (Cat. No. 03EX706), pp. 72–79. IEEE (2003)
14. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **6**(1), 58–73 (2002)
15. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360), pp. 69–73. IEEE (1998)
16. Suganthan, P.N.: Particle swarm optimiser with neighbourhood operator. In: *Proceedings of the 1999 Congress on Evolutionary Computation (CEC 1999)* (Cat. No. 99TH8406), vol. 3, pp. 1958–1962. IEEE (1999)
17. Howden, D.J.: Bushfire surveillance using dynamic priority maps and swarming unmanned aerial vehicles. Ph.D. thesis, Ph.D. dissertation, Swinburne University of Technology, Hawthorn, Victoria (2013)
18. Khan, A.: Coordinated unmanned aerial vehicles for surveillance of targets. Ph.D. thesis, Queen Mary University of London (2015)
19. Ravankar, A., Ravankar, A., Kobayashi, Y., Hoshino, Y., Peng, C.C.: Path smoothing techniques in robot navigation: state-of-the-art, current and future challenges. *Sensors* **18**(9), 3170 (2018)
20. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS 1995)*, pp. 39–43. IEEE (1995)
21. Olorunda, O., Engelbrecht, A.P.: Measuring exploration/exploitation in particle swarms using swarm diversity. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 1128–1134. IEEE (2008)