Cluster Computing (2021) 24:2495–2509
https://doi.org/10.1007/s10586-021-03276-6

# Real time UAV path planning by parallel grey wolf optimization with align coefficient on CAN bus

Vahid Jamshidi[1] · Vahab Nekoukar[2] · Mohammad Hossein Refan[3]

Received: 30 June 2020 / Revised: 26 January 2021 / Accepted: 24 March 2021 / Published online: 10 April 2021


**Abstract**
Unmanned aerial vehicle (UAV) path planning is a complex optimization problem, which aims to achieve an optimal or nearly optimal flight path despite various threats and constraints. In this paper, an improved version of Gray Wolf Optimization (GWO) is proposed to solve the UAV 3D path planning problem which considers the dynamics of the UAV. In improved GWO, a variable weighting called "align coefficient" is defined to deal with the problem of waypoint scattering. The parallel GWO is applied to reduce the computation time which makes the possibility of real-time implementation. Given the existence and unique features of CAN bus in UAVs, it is used as a platform to migrate individuals in the parallelization process. The simulation results demonstrate that applying improved GWO generates better performance for UAV 3D path planning problems compared to the conventional GWO, GA, PSO, SA, improved GA and improved PSO.

**Keywords** Grey wolf optimization · UAV Path planning · CAN bus · Parallelization · Multi-master model

## 1 Introduction

The increasing of UAV reliability and technical capabilities in operational missions causes dramatic growing of the UAV industries. Path planning is one of the critical technologies in UAVs that tries to offer an optimal path from origin to destination which consideration of various threats and constraints in the flight field and capabilities of the UAV. Intelligent path planning can be used to extend the lifetime of UAVs, which has attracted the attention of many researchers in recent years [1].

Metaheuristic optimization techniques have become an alternatives of optimization method in several engineering fields because of their excellent performances in real world optimization problems. Up to now, several metaheuristics have been proposed to solve the path planning problem. For example, in [2], artificial neural networks optimized by the imperialist competitive algorithm were used for UCAV's global path planning. In [3], ant colony was applied for UAV path planning. The authors of [4] proposed the particle swarm optimization (PSO) algorithm and the Bezier curve-based model for path planning. In [5] comprehensively improved PSO is applied for efficient UAV path planning. In [6], Genetic algorithm (GA) was used to find the optimal flight path for UAVs in a 3D environment. In [7] novel SA algorithm applied for Balanced UAV Task Assignment and Path Planning. In [8] an improved bat algorithm is applied for 3D path planning for UCAV. In [9] hybrid meta-heuristic DE/BBO algorithm is applied for UCAV path planning. In [10] a modified firefly algorithm is used for UCAV path planning. In [11] a Bat Algorithm with mutation is applied for UCAV Path Planning In [12] A hybrid meta-heuristic DE/CS algorithm is applied for UCAV path planning. In [13] A hybrid meta-heuristic DE/CS algorithm is applied for 3D path planning

✉ Vahab Nekoukar
v.nekoukar@sru.ac.ir

Vahid Jamshidi
vahid_mordad200020001@yahoo.com

Mohammad Hossein Refan
refan@sru.ac.ir

1 Electrical Engineering School, Shahid Rajaee Teacher Training University, Tehran, Iran

2 Electrical Engineering School, Shahid Rajaee Teacher Training University, Tehran, Iran

3 Electrical Engineering School, Shahid Rajaee Teacher Training University, Tehran, Iran

for UCAV. And in [14] Bi-Directional Adaptive A* Algorithm is used for Optimal UAV Path Planning in Large-Scale and Under Multi-Constraints.

Gray wolf optimization (GWO) is one of the meta-heuristic algorithms that considers the searching, hunting behavior, and the social hierarchy of the grey wolves [15]. The GWO algorithm has few parameters, and having a simple structure, flexibility and avoid falling into a local minimum, is one of its highlights [16]. Due to less randomness and different numbers of individuals in local and global search procedures, the GWO algorithm is faster to converge and easier to use [17]. The GWO algorithm has been shown to be more efficient than the other bionic algorithms [17]. In [18], the Grey wolf optimization was used for unmanned combat aerial vehicle path planning. In [19], a novel hybrid GWO algorithm is used for UAV path planning. In this article, the exploration and exploitation capabilities are combined and the convergence analysis of the HSGWO-MSOS algorithm is presented based on the linear difference equation. Ref. [20] developed a novel hybrid PSO–GWO method to solve unit commitment problem. Ref. [21] uses a PSO–GWO algorithm for autonomous guided robot. In this article, the proposed algorithm consists of three steps: (1) hybridization of the GWO-PSO algorithm. (2) all optimal and feasible points are integrated with Local Search technique. (3) collision avoidance and detection algorithm. In [22], GWO algorithm is used for UAV path planning and the performance of GWO is compared with A*, D*, IBA, BBO, PSO, GSO, WOA and SCA, so as to find the optimal method. The results of this article show that the GWO algorithm outperforms the other deterministic and meta-heuristic algorithms in path planning for 3D multi-UAV. In [23], an improved GWO algorithm is designed for solving the objective model. In this article, Time and space constraints are intended to deal with the complexity of the environment. In this article, the idea of PSO is used in order to memorize the optimal solution of the particle motion history into the GWO algorithm.

Many studies of UAV path planning have evaluated their proposed methods in the simulation. There are significant challenges to using microcontrollers in real-time applications [24]. For example, the GWO algorithm, like other metaheuristics, requires time-consuming computation, while microcontrollers are low-resource platforms. To deal with this constraint, parallelization can be used to increase computational speed by dividing tasks into multiple microcontrollers to solve complex and time-consuming problems [25].

Some metaheuristics such as parallel GA and parallel PSO have been successfully applied to a wide range of practical problems such as UAV path planning. In [26], a parallel GA was used for optimal cooperative path planning of UAVs. In [27], a comparison of parallel genetic algorithm and particle optimization for real-time UAV path planning was presented. However, few studies have been conducted to solve the engineering problems by the parallel GWO algorithm. To implement the parallel GWO algorithm process in systems that have multiple independent processors, an interface for migration and communication is needed.

Controller area network (CAN) is an open protocol designed for reliable communication in automotive, aerospace and robotic applications. CAN has developed on UAVs. With the increasing growth of industrial optimization problems (such as UAV path planning) and their high computations on the one hand and the existence of CAN bus in these systems on the other hand, caused this bus to be used for parallelization and distribution of computations. Therefore, it is a potential platform for implementing the parallel algorithms on UAVs. In addition, this bus has unique features that make it suitable for migration in metaheuristics, including broadcast message, automatic retransmission, Low cost, small frame, and a multi-master structure. In [24], we introduced a multi-master structure for parallelization of the GA and PSO for UAV path planning on CAN bus.

In this paper, for the first time the problem of UAV path planning with multi-master and parallel GWO algorithm is performed on CAN bus. We compared the results of the UAV path planning by parallel GWO algorithm with the results of the parallel GA and the parallel PSO algorithms presented in [24] and analyzed the network performance. It also improved the performance of the GWO algorithm in the face of UAV path planning by defining the “alignment path” and modifying this algorithm with the “align coefficient”. The advantage of this coefficient is to overcome the problem of waypoint scattering in the UAV path planning. Finally, to evaluate the proposed method, an improved GWO algorithm with different number of nodes on CAN bus is implemented. Simulation are carried out, and comparisons are made. Results show that the improved GWO algorithm with align coefficient works better than the conventional GWO, GA, PSO, SA, improved GA and improved PSO.

The main contribution of this manuscript is the presentation and analysis of the UAV path planning problem by the parallel, multi-master and improved GWO algorithm on the industrial CAN bus. In this way, an industrial solution is provided that is effective, inexpensive and can be implemented in existing microcontrollers and hardware. Compared to studies, we present the following innovations in this article.

(1) **CAN bus:** In this paper, for the first time the problem of UAV path planning with GWO algorithm is performed on CAN bus.
(2) **Parallel:** We have used the parallel GWO algorithm with alpha sharing to improve performance for the UAV path planning problem.
(3) **multi-master structure and Asynchronous:** In this paper, a communication strategy for the parallelized GWO is proposed for implementation in an industrial system. The multi-master structure with asynchronous properties has been analyzed in the Parallel GWO algorithm.
(4) **align coefficient:** It also improved the performance of the GWO algorithm in the face of UAV path planning by defining the "alignment path" and modifying this algorithm with the "align coefficient".

The remainder of this paper is organized into five sections. Section 2 presents the mathematical model of the problem, including the digital terrain, the cost function, conventional GWO and multi-master structure of CAN bus. Section 3 presents the method of parallel GWO on CAN bus for UAV path planning. And finally, Section 4 presents the evaluation and results.

## 2 Mathematical model of the problem

### 2.1 Terrain modeling and characteristics of the UAV

In path planning studies, it is assumed that terrain constraint is known [28]. One of the issues in 3D environments is how to obtain ground elevation. We generated a 2D matrix to represent ground. The value of each member of the matrix indicates the height of that area from the ground [27]. Radar threat is one of the potential threats in military UAVs. Here a 3D cylinder model in space is defined as a threat area [29] with coordinates $(x_i.y_i.d_i)$, where $(x_i.y_i)$ is the center of the $i$ th radar on the XOY plain and $(d_i)$ is the diameter of $i$ th radar.

Due to the complexity of the path planning problem, waypoints can be used as a suitable strategy. An UAV can split the path to several waypoints and passing through them toward the destination. In fact, the path planning problem is to find the optimal coordinates for these waypoints so that the UAV crosses them at the least cost to reach the destination. In [30], autonomous waypoints planning were used for multi-rotor UAVs trajectory generation. In [31], the waypoint guidance was used for autonomous underwater vehicle path planning. A path consists of several connected waypoints with coordinates $(x_i.y_i.z_i)$. Of course, elevation of the trajectory at any waypoint $(z_i)$ is the same as the elevation of the environment in that cell, and UAV will fly slightly above the ground. The number of waypoints is constant and equal to 7. This number is relevant to the problem and all algorithms are evaluated with 7 waypoints. Of course, it is possible to select two or more waypoints with the same coordinates. This means that, for example, if two consecutive waypoints have the same coordinates, only one of them is considered in the calculations. Experience has shown that this number of waypoints is appropriate for the environment intended for this document. A 3D visualization of the flight path, terrain, and threat areas is shown in Fig. 1.

A mathematical model of UAV should be applied to solve path planning problem. The model explains relations between UAV movement, speed, fuel, power, and etc. In this paper, the model of CP-1 airplane is used for modeling the vehicle. The CP-1 model was described in [27] based on [32], assuming that UAV flies in three directions: moving horizontally at a constant speed, climbing by a constant angle, and descending.

### 2.2 Cost function

In general, there are several constraints in UAV path planning, including: path length $(C_l)$, path altitude $(C_a)$, avoiding the radar areas $(C_d)$, avoid of exceeding the maximum power of the UAV $(C_p)$, Avoid choosing paths that run out of fuel before reaching the destination $(C_f)$ and finally, avoiding collision with the ground $(C_c)$. These constraints are described in Eq (1).

$$F_{\cos t} = 10 \times \left(C_l + C_a + C_d + C_p + C_f + C_c\right) \tag{1}$$

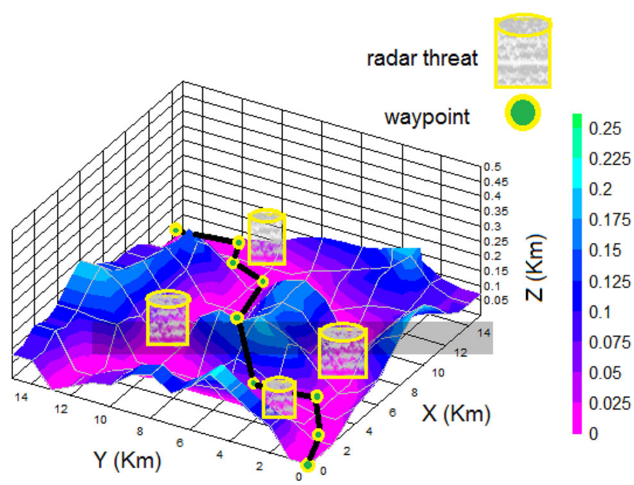



**Fig. 1** 3D visualization of the flight path and threat areas (in area of $256\text{km}^2$)

$C_l$, $C_a$ and $C_d$ are optimization criteria that are used to improve the quality of path. These three criteria are set within the range of $[0, 1]$. $C_c$, $C_f$, and $C_p$ are feasibility criteria that are considered as a penalty function. In valid paths, the cost of these criteria is zero. Paths violating these criteria are not valid. The invalid paths are those that the output of one or more of the feasibility criteria is in range of $[3+0, 3+1]$. Value 3 is considered as a penalty, helping to distinguish feasible solutions from other ones. Thus, the sum of the feasible path is less than or equal to 3. Since the output of the cost function is integer, coefficient 10 is used to increase the accuracy of the path evaluation. Given the coefficient 10 in the cost function, the cost of valid paths is between zero and thirty, and the cost of invalid paths is greater than thirty. All invalid paths have a higher value of the cost compared with the valid ones [24, 27, 32], (Table 1). The optimization algorithm in the search process seeks to minimize the cost function. This cost function is extremely complex and demonstrates the quality of our path planning algorithm. the complexity analysis in this document is described in Eq. (2).

$$\mathbf{O}((\text{Number of iterations})(\text{Number of gray wolves})(\text{Number of waypoints in each node})(\text{waypoint size})) \tag{2}$$

## 2.3 Conventional GWO algorithm

The GWO algorithm was developed by Mirjalili and Lewis [15]. In this algorithm, the population was divided into four groups. It is assumed that the initial three best solutions are alpha (α), beta (β) and delta (δ) respectively and the other solutions are omega (ω). In the GWO algorithm, α, β and δ have sufficient knowledge of the potential prey position. During the search and hunting, all omega are driven by these three gray wolves based on Eqs. (3–9).

$$\vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{P}_\alpha - \vec{P}\right| \tag{3}$$

$$\vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{P}_\beta - \vec{P}\right| \tag{4}$$

$$\vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{P}_\delta - \vec{P}\right| \tag{5}$$

$$\vec{P}_1 = \vec{P}_\alpha - \vec{A}_1 \cdot \left(\vec{D}_\alpha\right) \tag{6}$$

$$\vec{P}_2 = \vec{P}_\beta - \vec{A}_3 \cdot \left(\vec{D}_\beta\right) \tag{7}$$

$$\vec{P}_3 = \vec{P}_\delta - \vec{A}_1 \cdot \left(\vec{D}_\delta\right) \tag{8}$$

$$\vec{P}(t+1) = \frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3} \tag{9}$$

Where $\vec{P}_\alpha$ is the position of the alpha vector, $\vec{P}_\beta$ is the position of the beta vector, $\vec{P}_\delta$ is the position of the delta vector, $\vec{P}$ is the position vector of the current solution. Using the above equations, the grey wolf could update its position based on the prey position. $\vec{C}$ and $\vec{A}$ are coefficient vectors.$\vec{r}_1, \vec{r}_2$ are random numbers in [0, 1], and $\alpha$ is gradually decreased from 2 to 0,

**Table 1** Cost function of uav path planning

| constrain | Equation | | range |
|---|---|---|---|
| Length | $C_l = 1 - \left(\frac{L_D}{L}\right)$ | $L$: the length of the generated path<br>$L_D$: length of a direct path between an origin and destination | $[0, 1]$ |
| Altitude | $C_a = \frac{Z_{mean}}{Z_{max}}$ | $Z_{mean}$: the average altitude of the generated path<br>$Z_{max}$: maximum altitude of the ground | $[0, 1]$ |
| Radar | $C_d = \frac{\sum_i \sum_j L_{dij}}{\sum_{i=1}^{n} d_i}$ | $L_{dij}$: length of $i$ th sub-path that UAV passes through $j$ th radar<br>$d_i$: diameter of $i$ th radar | $[0, 1]$ |
| Ground collision | $C_c = \begin{cases} 0 \to L_c = 0 \\ 3 + (\frac{L_c}{L}) \to L_c > 0 \end{cases}$ | $L_c$: The total length of the generated path that UAV moves below the ground level | 0 or $[3, 4]$ |
| power | $C_p = \begin{cases} 0 \to L_{not} = 0 \\ 3 + (\frac{L_{not}}{L}) \to L_{not} > 0 \end{cases}$ | $L_{not}$: The sum of the length of the generated path that need more power than the UAV's available power | 0 or $[3, 4]$ |
| Fuel | $C_f = \begin{cases} 0 \to F_t \le F_i \\ 3 + (1 - \frac{F_s}{F_t}) \to F_t > F_i \end{cases}$ | $F_s$: fuel required for direct flights from the source to the target,<br>$F_i$: initial UAV fuel level<br>$F_t$: quantity of fuel necessary for the generated path | 0 or $[3, 4]$ |

$$\vec{A} = 2\,\vec{\alpha}\,.\,\vec{r_1} - \vec{\alpha} \tag{10}$$

$$\vec{C} = 2.\,\vec{r_1} \tag{11}$$

The flowchart of the conventional GWO algorithm is presented in *MERGEFORMAT Fig. 2.

### 2.4 CAN bus and multi-master structure

CAN bus is a serial communication protocol that enables microcontrollers to communicate with each other without a host computer. It was originally designed for multiplex electrical wiring in the cars, but is now widely used in UAVs.

CAN uses a media access method called "Carrier Sense Multiple Access with Arbitration on Message Priority" [33]. With this method, CAN data link layer provides multi-master capabilities. This means that if the bus is idle, all nodes are allowed to access the bus. If multiple nodes want to communicate at the same time, the message with the highest priority wins the arbitration of the bus and gets the right of transmission. The system designer sets a priority for each message. The ID (identifier) is the part of the message frame that shows the priority. The value of "0" is the highest priority [34].

For example, consider three nodes at the same time transmit their messages with IDs 110010111XX (node 1), 110011XXXXX (node 2) and 11001011001 (Node3) on the bus. Nodes start their frames with the "start bit" and then transmit its first five bits (bit 10 to bit 6) of their ID. Due to the identically of these bits, arbitration is not performed. In the 6th ID bit (bit 5), node 2 transmits the value 1 (recessive) and nodes 1 and 3 transmit the value 0 (dominant). At this time, the receiver pin (RX) of the node 2 sees the value of 0, while the transmitter pin (TX) transmits the value of 1. Therefore, node 2 realizes that there is a collision and it lost arbitration. Likewise, node 1 stops the transmitting (in bit 1 of ID) which allows node 3 to continue its transmission without any loss of data. The node with the lowest ID has the highest priority, and therefore always wins the arbitration (Fig. 3).

Given the features of the CAN bus, including those discussed above, this protocol is very suitable for implementing of parallel metaheuristics. In addition, the CAN is available in cars and UAVs and thus can be adapted for various needs including path planning without imposing additional cost and space requirements.

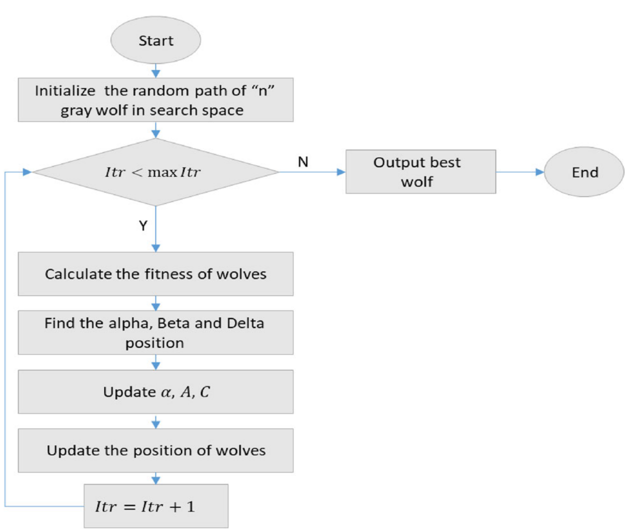



**Fig. 2** Flowchart of the conventional GWO algorithm.

## 3 Method of path planning

### 3.1 Parallel GWO UAV path planning on CAN bus

In this paper, GWO is used to find optimal path of UAV. Each solution is introduced as a wolf position, which is a string of binary values. The population of GWO consists of 240 wolves. Each wolf position represents a path from an origin to a destination. Each path consists of seven waypoints, an origin, and a destination that are connected to each another. Therefore, each wolf vector contains the coordinates of seven waypoints. Each waypoint represents in 8 bits. These coordinates include the position of the X-axis (in 4 bits) as well as the position of Y-axis (in 4 bits) of each waypoint in the matrix of the ground. The height (Z-axis) of any waypoint is considered equal to the height of ground at that waypoint, and UAV will fly slightly above the ground. So, each wolf vector consists of 56 bits. GWO algorithm begins by random creation of an initial path of gray wolves. The initial population (seven waypoints in each path) is randomly selected in the 3D search space, regardless of the feasibility of the solutions. Each path has a fitness value that compares the quality of this particular solution with other paths in the population.

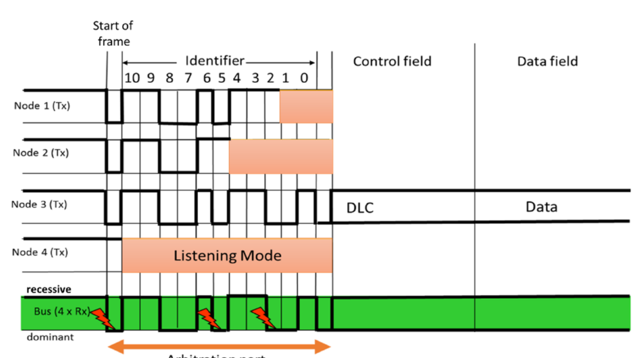



**Fig. 3** Non distractive arbitration mechanism in CAN bus

In the parallel GWO algorithm, the wolves' population is divided into several subpopulations, each being processed in a single node. To improve performance, wolves can migrate between nodes during the parallelization process. In the parallel GWO on the CAN bus, wolves migrate on a common bus. The structure of the migration between nodes is multi-master and each node is known as a master in the network. Each master executes the GWO algorithm in its own subpopulation and identifies alpha, beta, and delta in each iteration. Other wolves are also known as omega and are driven by these three wolves.

To prevent premature convergence, data exchange between nodes occurs at every 100 iterations. So, when each master completes the path planning process in one hundred iterations, it transmits its best individual, alpha, on the bus. All nodes can view and use all of the messages sent on the bus. Depending on the ID assigned to each message, the nodes place the messages in the received buffers and raise the flag.

In each iteration of the GWO algorithm, each master checks their receiving buffers. If there is a new message, it would compare that message (which is actually the alpha coordinates of another node) to one of its population members. If the received message is better, the master replaces it, lowers the flag and continues the algorithm. The flowchart of UAV path planning by presented parallel GWO on CAN bus is shown in Fig. 4. According to the non-destructive arbitration mechanism in the CAN bus, messages are not destroyed. Whenever the path planning process is completed in 100 iterations per node, it sends the alpha coordinates to the bus regardless of the status of the other master. Therefore, the asynchronous algorithm can be implemented in this network because it is more flexible and can maximize efficiency.

To transmit the alpha on CAN bus, the message frame is configured as Fig. 5. We used 11-bit ID and 8-byte data. An array is used to represent data bytes. The 56 bits of data field belong to the alpha coordinates and the reminder belongs to the alpha fitness value.

### 3.2 Improved GWO UAV path planning with "align coefficient"

Population of gray wolves are randomly, without any restrictions, initialized all through the space domain. As mentioned, each wolf contains a vector of waypoints. Given that the waypoints are selected without restriction in the search space, one of the major problems in finding the optimal path in a short time is the scattering of the waypoints. As the number of waypoints increases, the complexity of the problem increases, and the algorithm must make every effort to align these waypoints from source to destination. Thus, although alpha is the nearest wolf to the prey at the beginning of the search, it may be far from the end result, let the alone other wolves. To deal with this

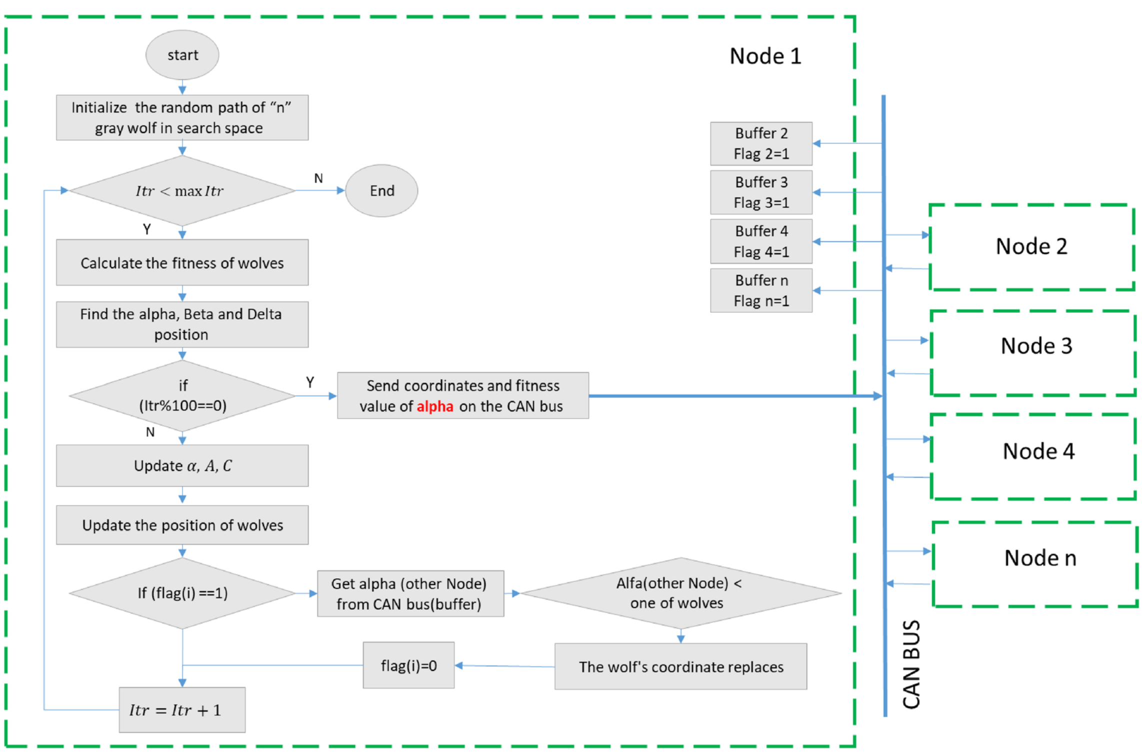



**Fig. 4** Parallel GWO on CAN flowchart

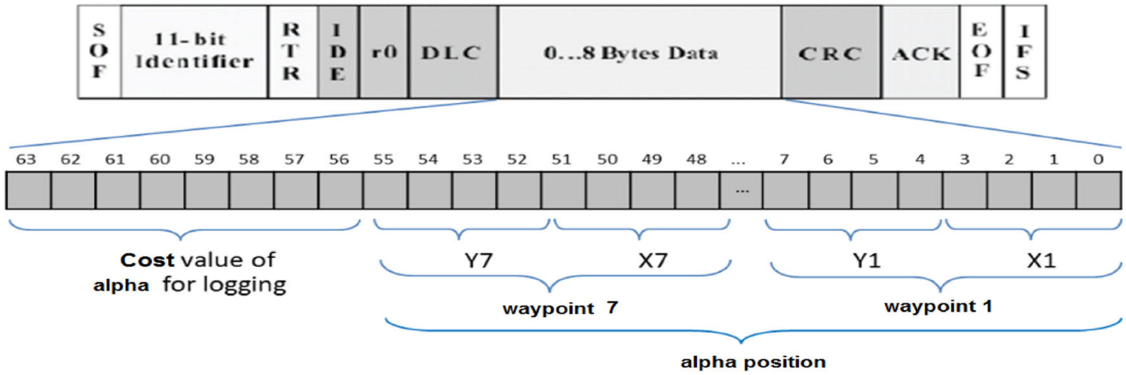



**Fig. 5** CAN frame

problem, we define the "alignment path" (Fig. 6). The following steps are taken to this end.

(1) Draw a straight line from the source to the destination.

(2) We assume that the number of waypoints is seven. Divide the drawn line into several equal sections. The seven waypoints are located at the intersections of this line, respectively.

(3) The position of the waypoints on the line forms a vector. In fact, it forms a potential path for the UAV. This potential path is called the "alignment path".

(4) If the wolves (paths) have a relative alignment at the beginning of the search, they are more likely to find the optimal path. Therefore, if the vectors of the scattered paths are inclined toward the vector of the "alignment path", the likelihood of alignment of the paths is increased and thus the search process improves.

In the conventional wolf algorithm, gray wolves perform the search process according to the position of alpha, beta and delta. In fact, the gray wolves move to the hypothetical prey position. Like wolves, the prey position is a vector. For example, in (Fig. 7a) the first member of the path vector moves to the first member of the hunting vector. Given that the paths are likely to be scattered at the beginning of the search. The method of updating positions should not only depend on the position of alpha, beta and delta (prey). Rather, it should be oriented towards the

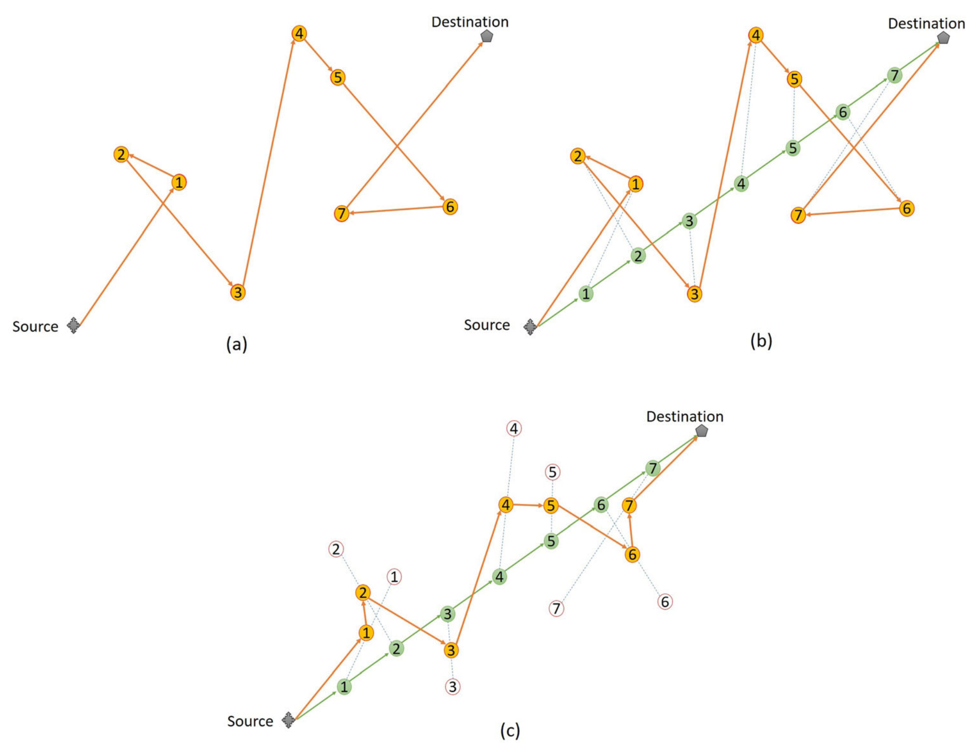



**Fig. 6** **a** A path with scattered waypoints **b** Added alignment path (green). **c** Move the scattered path vector to the alignment path vector

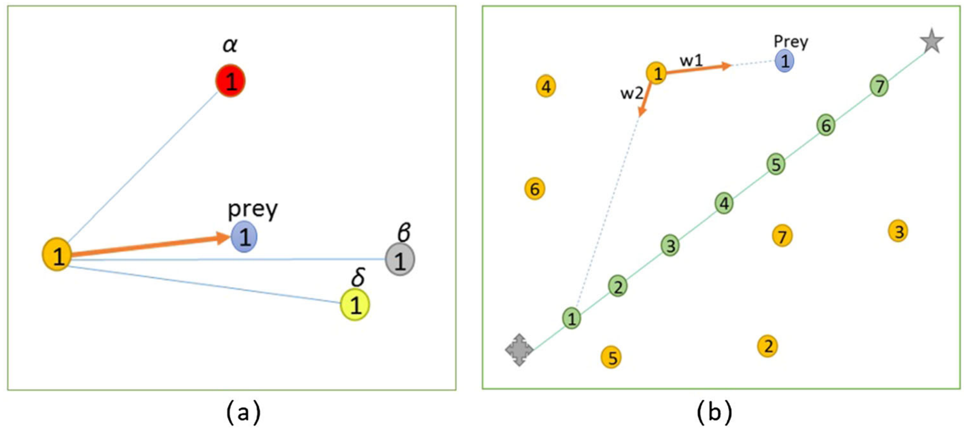



**Fig. 7** **a** conventional GWO, **b** drive the omega wolves with the prey position and the path alignment

"alignment path". When the search begins, each of the waypoints, in addition to the prey, moves along the corresponding waypoint in the "alignment path" (Fig. 7b).

During the search and hunting, all the omegas, in addition to the three gray wolves (Alpha, Beta and Delta), are also driven by the "alignment path" based on Eqs. (12–14).

$$\vec{D}_{align} = \left| \vec{C}_4 \cdot \vec{P}_{align} - \vec{P} \right| \tag{12}$$

$$\vec{P}_4 = \vec{P}_{align} - \vec{A}_4 \cdot \left( \vec{D}_{align} \right) \tag{13}$$

$$\vec{P}(t+1) = w_1 \left( \frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3} \right) + w_2 \vec{P}_4 \tag{14}$$

where $\vec{P}_{align}$ denotes the vector position of the "alignment path". $w_1$ denotes the "prey coefficient" and $w_2$ denotes the "align coefficient". In general, a large value for $w_2$ leads to faster convergence, which is useful in the early stage of convergence for the wolves' vector, but in the next period, large value for $w_2$ cause the wolves to miss the prey position and increased the probability of getting stuck in local optima. Therefore, the search coefficients were adjusted by the Eq. (15).

$$w_2 = \left( \frac{1 - \frac{itr}{\max itr}}{4} \right)^2 \qquad w_1 = 1 - w_2 \tag{15}$$

where "$itr$" is the iteration number and "max$itr$" is the termination number. $w_2$ is gradually decreased from ($0.25^2$) to 0.

## 4 Simulation-based evaluation

### 4.1 Evaluation and analysis of parallel GWO on CAN bus

To evaluate the performance of the proposed structure, the UAV path planning problem by parallel GWO is simulated on CAN bus. For this purpose, a population of 240 wolves (each of which is the vector of the seven waypoints) is considered. To parallelize this population, it is split between nodes connected to the bus. Each master executes the algorithm in a population of $240/n$, where $n$ is the number of nodes. Simulations are performed in the "Bus master Node Simulator" software. It helps simulating, monitoring and analyzing CAN message on bus. The parameter values applied in our simulation of the UAV path planning by parallel GWO are listed in *MERGEFORMAT Table 2. The simulation results of parallel GWO on CAN bus, with the number of nodes, are presented in *MERGEFORMAT Table 3. To compare the performance of the parallel GWO algorithm in UAV path planning, we compare the results of parallel GWO with the results of the parallel improved PSO and parallel improved GA presented in [24]. Given that metaheuristics are non-deterministic, each test is repeated 20 times and the average results are obtained.

Where, the "Execution time" is the runtime of the UAV path planning problem in 1500 iterations on the CAN bus. So, the "average of execution time" is the average of execution time in 20 trials. The "average result" is the average cost of paths that generated in 20 trials. Feasible results are those that cost less than 30. So, "Average of feasible result" is the average cost of feasible generated paths in 20 trials. "Percentage of feasibility" is the ratio of feasible results ($F_{cost} < 30$) to all results (in 20 trials).

In *MERGEFORMAT Table 3, the most Percentage of feasibility are obtained by the improved GA with 5 nodes. After parallel improved GA, the best performance is

**Table 2** UAV path planning parameter values

| Number of Nodes on CAN bus | 1 Node | 2 Node | 3 Node | 4 Node | 5 Node |
|---|---|---|---|---|---|
| Dimensional of the earth matrix | 16 × 16 | | | | |
| Terrain resolution | 1000m × 1000m | | | | |
| Number of waypoints in each path | 7 waypoints | | | | |
| Number of iterations | 1500 | | | | |
| Number of gray wolves | 240 | | | | |
| Number of gray wolves in each Node | 240 | 120 | 80 | 60 | 48 |
| waypoint size | 8bit (X, Y) | | | | |
| The size of the CAN frame | 8 bytes | | | | |
| Baud rate | 500 kbps | | | | |
| Send frame on BUS (number of migration) | 15 | 30 | 45 | 60 | 75 |
| number of trials | 20 | | | | |
| Simulation software | BUSMASTER 3.1 | | | | |

**Table 3** Result of the parallel GWO, parallel improved GA and parallel improved PSO on CAN bus

| Algorithm | Number of bus nodes | Average of execution time(s) | Average results | Average of feasible results | Percentage of feasibility (%) |
|---|---|---|---|---|---|
| GWO | 1 Node | 54.65 | 23.55 | 9.20 | 50 |
| | 2 Node | 36.45 | 22.70 | 8.70 | 50 |
| | 3 Node | 27.90 | 20.10 | 7.90 | 55 |
| | 4 Node | 21.49 | 18.95 | 7.50 | 60 |
| | 5 Node | 17.94 | 14.15 | 6.93 | 75 |
| Improved GA [24] | 1 Node | 69.08 | 18.85 | 6.55 | 55 |
| | 2 Node | 36.34 | 20.45 | 4.44 | 45 |
| | 3 Node | 24.34 | 16.85 | 5.67 | 60 |
| | 4 Node | 21.81 | 14.10 | 5.64 | 70 |
| | 5 Node | 21.45 | 10.80 | 5.13 | 80 |
| ImprovedPSO [24] | 1 Node | 71.15 | 21.05 | 4.88 | 45 |
| | 2 Node | 44.18 | 19.20 | 5.20 | 50 |
| | 3 Node | 34.49 | 16.40 | 5.16 | 60 |
| | 4 Node | 25.95 | 15.95 | 4.41 | 60 |
| | 5 Node | 21.61 | 16.05 | 4.58 | 60 |

obtained in the parallel GWO with 5 nodes. In terms of the average execution time of the problem, the GWO algorithm with 5 nodes has the best performance (fastest). It is clear that as the number of nodes increases, the population to be evaluated in each node decreases. As a result, the execution time of each iteration and the execution time of the whole path planning process (in 1500 iteration) are reduced. Path planning performance is also significantly improved by parallelizing and exchanging node information in multi master structure. So that the average cost in the GWO algorithm decreases from 23.55 to 14.15 and the percentage of feasible answers has been increased from 50% to 75%.

Network statistics are used to analyze the bus performance, which displays the details of messages transmitted and received on the CAN bus. This information is generated by the "BUSMASTER Node Simulation". It also presented network load as bus traffic and message rates per second. In *MERGEFORMAT Table 4, the CAN bus statistics are presented for simulation of the parallel GWO. As mentioned, each node in the network sends its best member (for use by other nodes) to the bus once in every 100 iteration. This means that in 1500 iteration, 15 messages will be sent from each node on the bus. Increasing number of nodes in the network increase communication and traffic on the network. However, the results show that cost of the communication and bus traffic in the proposed structure is very low. This means that in a system that has a CAN network and in which the other communication is running, this structure can be easily used. And without

**Table 4** CAN bus statistics of UAV path planning by parallel GWO

| Number of node | Number of transmitted messages | Number of transmitted messages in seconds | Peak load in the bus | Average load in the bus |
|---|---|---|---|---|
| 1 Node | 15 | 0.27 | $3 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| 2 Node | 30 | 0.82 | $5 \times 10^{-4}$ | $2 \times 10^{-4}$ |
| 3 Node | 45 | 1.61 | $8 \times 10^{-4}$ | $4 \times 10^{-4}$ |
| 4 Node | 60 | 2.79 | $10 \times 10^{-4}$ | $7 \times 10^{-4}$ |
| 5 Node | 75 | 4.18 | $13 \times 10^{-4}$ | $9 \times 10^{-4}$ |

adding new hardware, it solved the problem of parallel optimization in real time with the lowest communication cost.

The ratio between the average runtime of a node and the average runtime of m-node is called the "Speed UP". As shown in *MERGEFORMAT Fig. 8, as the number of nodes increases, the problem-solving speed improves and the Speed Up increases "near-linearly". This is very significant in industrial systems.

## 4.2 Evaluation and analysis of parallel improved GWO on CAN bus

In this section, the results of the simulation and analysis of the improved and parallel GWO algorithm are presented. The simulation results of parallel improved GWO on CAN bus, with the number of nodes, are presented in Table 5. Comparison of Table 5 with Table 3 indicates that the performance of the parallel GWO algorithm has been greatly improved by using the "align coefficient". So that in 90% of cases, in a network with 5 nodes, the result of the problem is feasible and the average cost in 20 iteration is less than 10 (9.15).

To reflect the good design of our GWO and cost function, we plot (in Figs. 9, 10, 11, 12, 13) the cost of the median, best, worse and average results (in 20 trials) generated by the parallel and improved GWO against time. At iteration 1, the average graph has a high cost (greater than 30) and therefore violates a few feasibility constraints. The optimization process rapidly converges and finds a first feasible solution and the algorithm continues until the maximum number of iterations is attained. Best Solution (blue dashed line) is the answer that ultimately reaches the lowest value in cost. Of course, in the early stages of searching, a solution may have a low cost, but then it falls into a local optimum and ultimately does not reach the desired answer. It is also possible that a solution in the early stages of the search will be costly and eventually close to an optimal value (Fig. 14).

As shown, by increasing the number of nodes in the parallel GWO, speed and performance of the path planning are improved. We achieved a computation time of 18.45s for 240 gray wolves and 1500 iteration (by 5 Node on CAN bus and in area of $256km^2$). This allows the simulation to generate very high-quality trajectories in a wide range of complex 3D environments and this time allows the UAV to compute the next path, while flying the current one. For example, if the speed of the UAV is 500 km/h, it will travel 2560 m in 18.5 s. Our GWO path planning algorithm can easily generate paths longer than this distance and thus allows computation while in flight.

Table 6 presents the simulation results of the UAV path planning problem in a network with 5 nodes by different algorithms. Comparison of Table 5 with Table 3 as well as the results of Table 6 indicates that the proposed GWO algorithm works better than the conventional GWO, GA, PSO, SA, improved GA and improved PSO.

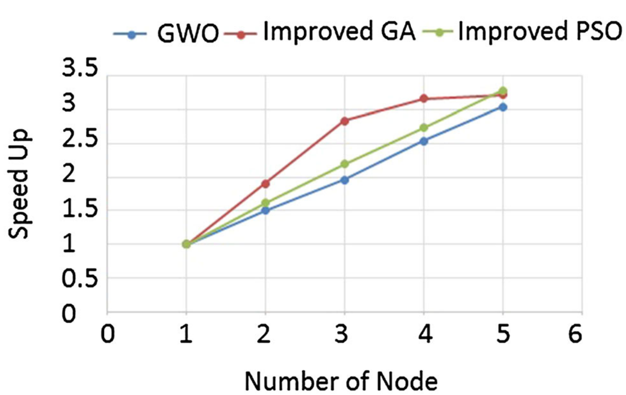



**Fig. 8** Speedup of parallel GWO, parallel improved PSO and parallel improved GA on CAN bus

# 5 Conclusion

UAV path planning is extremely complex when considering the real environment and dynamic constraints of the UAV. We used the parallel improved GWO algorithms to attack this complexity and produce solutions in a relatively

**Table 5** Result of the parallel improved GWO on CAN bus

| Algorithm | Number of bus nodes | Average of execution time (s) | Average results | Average of Feasible results | Percentage of feasibility (%) |
|---|---|---|---|---|---|
| Improved GWO | 1 Node | 59.10 | 15.70 | 5.08 | 65 |
| | 2 Node | 34.57 | 12.25 | 6.28 | 80 |
| | 3 Node | 26.46 | 11 | 5.06 | 80 |
| | 4 Node | 21.94 | 11.25 | 5.38 | 80 |
| | 5 Node | 18.45 | 9.15 | 5.94 | 90 |

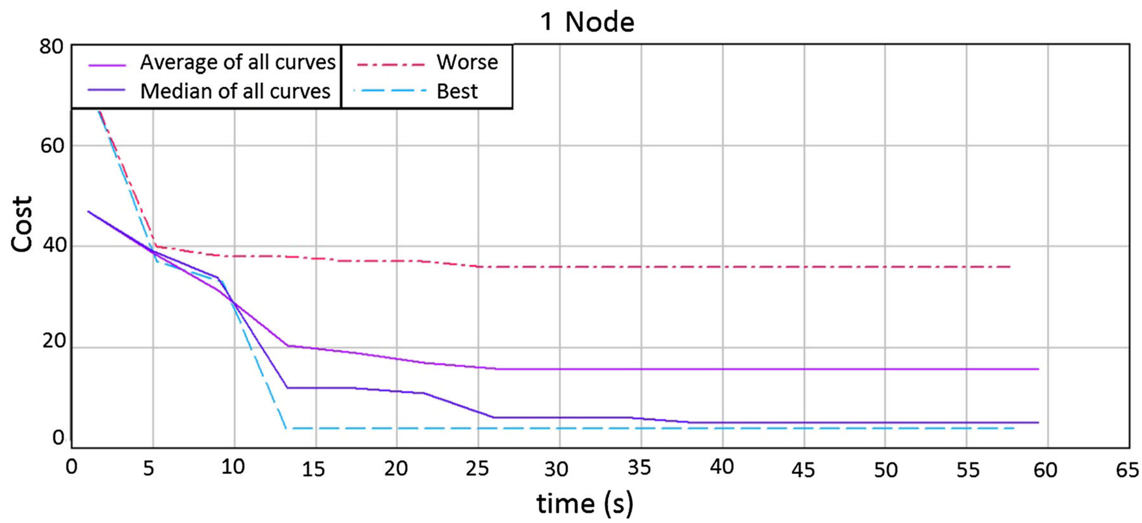
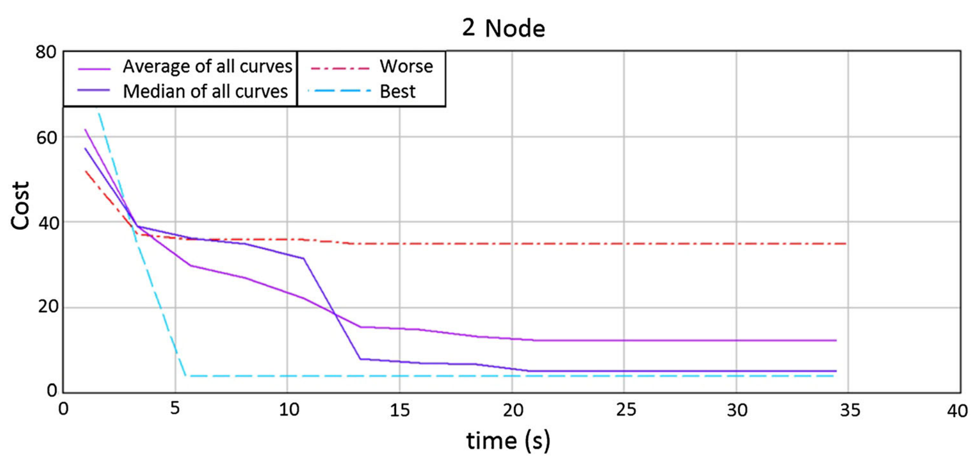



**Fig. 9** Cost of the solutions generated by improved GWO by 1 Node on CAN bus (in 20 traisl)

**Fig. 10** Cost of the solutions generated by improved GWO by 2 Node on CAN bus (in 20 trials)




short computation time. For this purpose, we applied the guidance waypoints and used the alignment coefficient to avoid the scattering of the waypoints. We used CAN's capacity to parallelize the GWO algorithm. The experimental results show that the improved parallel GWO algorithm can acquire an effective path successfully, and the comparative results also show the superiority of the proposed algorithm over the conventional GWO, GA, PSO, SA, improved GA and improved PSO in solving the UAV path planning problem. The proposed multi-master structure is applicable where it is necessary to optimize a real-time problem as well as where CAN is also available. In addition to the methods described in the paper, some of the most demonstrative computational algorithms can be used

**Fig. 11** Cost of the solutions generated by improved GWO by 3 Node on CAN bus (in 20 trials)

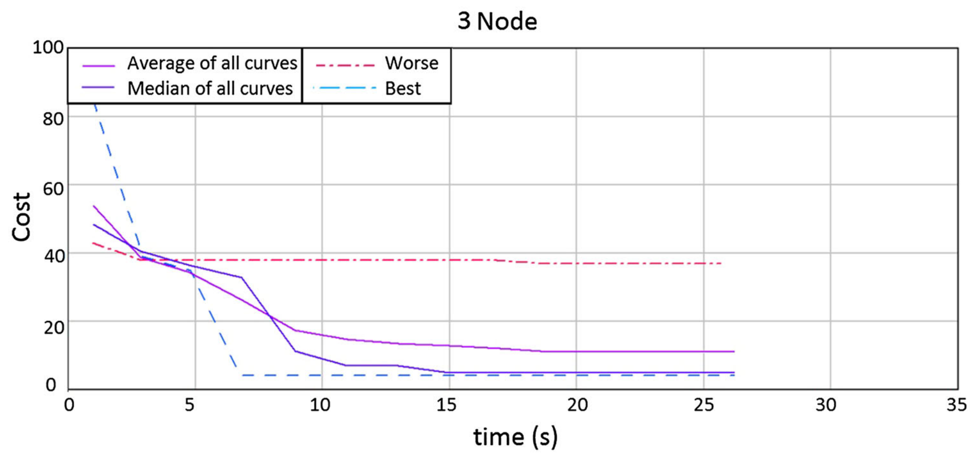



**Fig. 12** Cost of the solutions generated by improved GWO by 4 Node on CAN bus (in 20 trials)

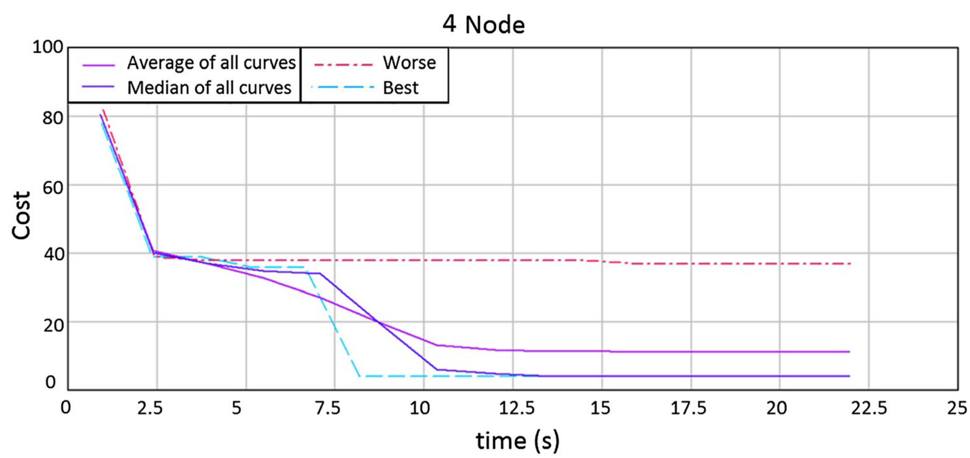



**Fig. 13** Cost of the solutions generated by improved GWO by 5 Node on CAN bus (in 20 trials)

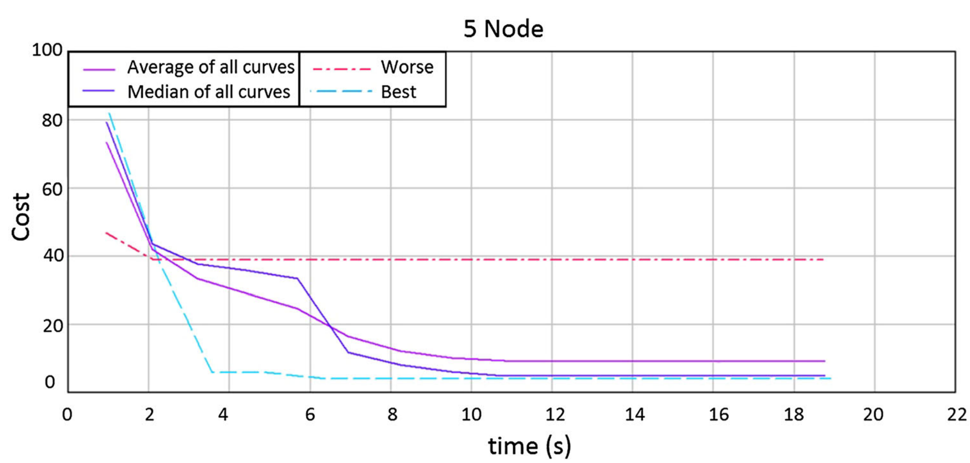


**Fig. 14** Comparison of the average results with the number of different nodes on CAN bus

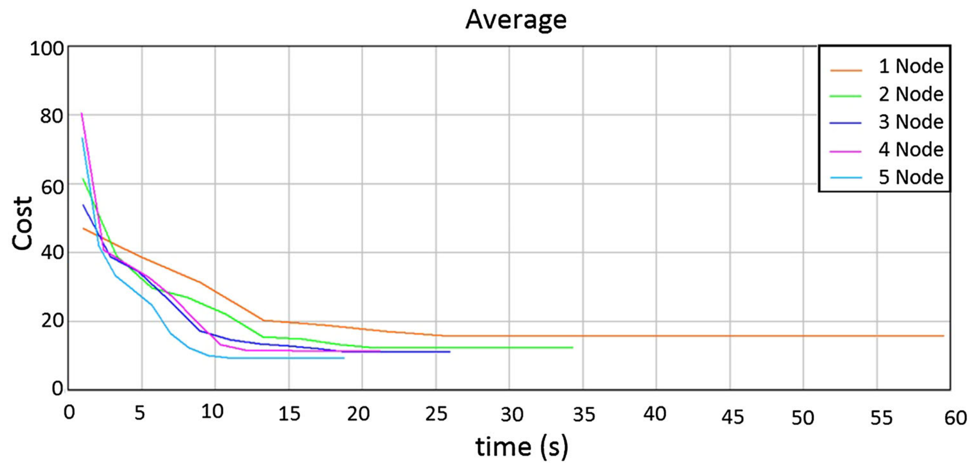



**Table 6** Comparision of parallel improved GWO by other algorithm on CAN bus

| Algorithm | Number of bus nodes | Average of execution time (s) | Average results | Average of feasible results | Percentage of feasibility (%) |
|---|---|---|---|---|---|
| SA | 5 Node | 48.24 | 24.65 | 6.43 | 35 |
| Basic GA | | 14.46 | 19.65 | 4.9 | 50 |
| Basic PSO | | 12.32 | 32.7 | 6 | 10 |
| GWO | | 17.94 | 14.15 | 6.93 | 75 |
| Improved GA | | 21.45 | 10.80 | 5.13 | 80 |
| Improved PSO | | 21.61 | 16.05 | 4.58 | 60 |
| Improved GWO | | 18.45 | 9.15 | 4.57 | 90 |

to solve problems, such as MBO, EWA, EHO, MS, SMA and HHO. For further work, these methods can be studied for UAV parallel path planning on the CAN bus.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References


1. Zhang, L., Liu, Z., Zhang, Y., Ai, J.: Intelligent Path Planning and Following for UAVs in Forest Surveillance and Fire Fighting Missions*. In: 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC). pp. 1–6 (2018)
2. Duan, H., Huang, L.: Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning. Neurocomputing **125**, 166–171 (2014). https://doi.org/10.1016/j.neucom.2012.09.039
3. Konatowski, S., Pawłowski, P.: Ant colony optimization algorithm for UAV path planning. In: 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET). pp. 177–182 (2018)
4. Tharwat, A., Elhoseny, M., Hassanien, A.E., Gabel, T., Kumar, A.: Intelligent bézier curve-based path planning model using chaotic particle swarm optimization algorithm. Clust. Comput. (2018). https://doi.org/10.1007/s10586-018-2360-3
5. Shao, S., Peng, Y., He, C., Du, Y.: Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. ISA Trans. **97**, 415–430 (2020). https://doi.org/10.1016/j.isatra.2019.08.018
6. Sonmez, A., Kocyigit, E., Kugu, E.: Optimal path planning for UAVs using Genetic Algorithm. In: 2015 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 50–55 (2015)
7. Huo, L., Zhu, J., Wu, G., Li, Z.: A novel simulated annealing based strategy for balanced UAV task assignment and path planning. Sensors. **20**, 4769 (2020). https://doi.org/10.3390/s20174769
8. Wang, G.-G., Chu, H.E., Mirjalili, S.: Three-dimensional path planning for UCAV using an improved bat algorithm. Aerosp. Sci. Technol. **49**, 231–238 (2016). https://doi.org/10.1016/j.ast.2015.11.040
9. Wang, G., Guo, L., Duan, H., Liu, L., Wang, H., Shao, M.: Path planning for uninhabited combat aerial vehicle using hybrid meta-heuristic DE/BBO algorithm. Adv. Sci. Eng. Med. **4**, 550–564 (2012). https://doi.org/10.1166/asem.2012.1223
10. Wang, G., Guo, L., Duan, H., Liu, L., Wang, H.: A modified firefly algorithm for UCAV path planning. Int. J. Hybrid Inf. Technol. 123–144 (2012)
11. Wang, G., Guo, L., Duan, H., Liu, L., Wang, H.: A bat algorithm with mutation for UCAV path planning. https://www.hindawi.com/journals/tswj/2012/418946/

12. Wang, G.-G., Guo, L., Duan, H., Liu, L., Wang, H., Wang, J.: A hybrid meta-heuristic DE/CS algorithm for UCAV path planning. J. Inf. Comput. Sci. **9**, 4811–4818 (2012)
13. Wang, G.-G., Guo, L., Duan, H., Wang, H., Liu, L., Shao, M.: A hybrid metaheuristic DE/CS algorithm for UCAV three-dimension path planning. Sci. World J. **2012**, 583973 (2012). https://doi.org/10.1100/2012/583973
14. Wu, X., Xu, L., Zhen, R., Wu, X.: Bi-directional adaptive A* algorithm toward optimal path planning for large-scale UAV under multi-constraints. IEEE Access. **8**, 85431–85440 (2020). https://doi.org/10.1109/ACCESS.2020.2990153
15. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014). https://doi.org/10.1016/j.advengsoft.2013.12.007
16. Mirjalili, S.: How effective is the Grey Wolf optimizer in training multi-layer perceptrons. Appl. Intell. **43**, 150–161 (2015). https://doi.org/10.1007/s10489-014-0645-7
17. Gao, Z.-M., Zhao, J.: An improved grey wolf optimization algorithm with variable weights. Comput. Intell. Neurosci. **2019**, 2981282 (2019). https://doi.org/10.1155/2019/2981282
18. Zhang, S., Zhou, Y., Li, Z., Pan, W.: Grey wolf optimizer for unmanned combat aerial vehicle path planning. Adv. Eng. Softw. **99**, 121–136 (2016). https://doi.org/10.1016/j.advengsoft.2016.05.015
19. Qu, C., Gai, W., Zhang, J., Zhong, M.: A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. Knowl. Based Syst. **194**, 105530 (2020). https://doi.org/10.1016/j.knosys.2020.105530
20. Kamboj, V.K.: A novel hybrid PSO–GWO approach for unit commitment problem. Neural Comput. Appl. **27**, 1643–1655 (2016). https://doi.org/10.1007/s00521-015-1962-4
21. Gul, F., Rahiman, W., Alhady, S.S.N., Ali, A., Mir, I., Jalil, A.: Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO–GWO optimization algorithm with evolutionary programming. J. Ambient Intell. Humaniz. Comput. (2020). https://doi.org/10.1007/s12652-020-02514-w
22. Dewangan, R.K., Shukla, A., Godfrey, W.W.: Three dimensional path planning using Grey wolf optimizer for UAVs. Appl. Intell. **49**, 2201–2217 (2019). https://doi.org/10.1007/s10489-018-1384-y
23. Xu, C., Xu, M., Yin, C.: Optimized multi-UAV cooperative path planning under the complex confrontation environment. Comput. Commun. **162**, 196–203 (2020). https://doi.org/10.1016/j.comcom.2020.04.050
24. Jamshidi, V., Nekoukar, V., Refan, M.H.: Analysis of parallel genetic algorithm and parallel particle swarm optimization algorithm uav path planning on controller area network. J. Control Autom. Electr. Syst. **31**, 129–140 (2020). https://doi.org/10.1007/s40313-019-00549-9
25. Krishnan, P.S., Tiong, S.K., Koh, J.: Parallel distributed genetic algorithm development based on microcontrollers framework. In: 2008 First International Conference on Distributed Framework and Applications. pp. 35–40 (2008)
26. Shorakaei, H., Vahdani, M., Imani, B., Gholami, A.: Optimal cooperative path planning of unmanned aerial vehicles by a parallel genetic algorithm. Robotica. **34**, 823–836 (2016). https://doi.org/10.1017/S0263574714001878
27. Roberge, V., Tarbouchi, M., Labonte, G.: Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans. Ind. Inform. **9**, 132–141 (2013). https://doi.org/10.1109/TII.2012.2198665
28. Liu, Y., Zhang, X., Guan, X., Delahaye, D.: Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization. Aerosp. Sci. Technol. **58**, 92–102 (2016). https://doi.org/10.1016/j.ast.2016.08.017
29. Özalp, N., Sahingoz, O.K.: Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms. In: 2013 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 308–317 (2013)
30. Li, Y., Eslamiat, H., Wang, N., Zhao, Z., Sanyal, A.K., Qiu, Q.: Autonomous waypoints planning and trajectory generation for multi-rotor UAVs. In: Proceedings of the Workshop on Design Automation for CPS and IoT. pp. 31–40. Association for Computing Machinery, Montreal, Quebec, Canada (2019)
31. Ataei, M., Yousefi-Koma, A.: Three-dimensional optimal path planning for waypoint guidance of an autonomous underwater vehicle. Robot. Auton. Syst. **67**, 23–32 (2015). https://doi.org/10.1016/j.robot.2014.10.007
32. Anderson, J.D.: Introduction to flight. McGraw-Hill, New York (2012)
33. Aradi, S., Bécsi, T., Gáspár, P.: Development of vehicle on-board communication system for harsh environment. Acta Tech. Jaurinensis. **6**, 53–63 (2013)
34. Robert Bosch GmbH: CAN Specification Version 2.0, (1991)




**Vahid Jamshidi** is a PHD candidate in electronic engineering at Shahid Rejaee Teacher Training University. His main research interests are industrial communication and Evolutionary Algorithms.

**Vahab Nekoukar** received the B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the Khaje Nasir Toosi University in 2005, Tarbiat Modares University in 2007 and Iran University of Science and Technology in 2012, respectively. In 2014, he joined the School of Electrical Engineering, Shahid Rajaee Teacher Training University, as an assistant professor. His current research interests include control of biological systems, robotics, navigation and guidance systems.

**Mohammad Hossein Refan** received his B.Sc. in Electronics Engineering from Iran University of Science and Technology, Tehran, Iran in 1972. After 12 years working and experience in the industry, he started studying again in 1989 and received his M.Sc. and Ph.D. in the same field and the same University in 1992 and 1999 respectively. He is currently an Associate Professor of Electrical and Computer Engineering Faculty, Shahid Rajaee Teacher Training University, Tehran, Iran. He is the author of about 50 scientific publications on journals and international conferences. His research interests include GPS, DCS, and Automation System.