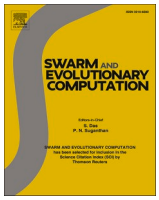
Swarm and Evolutionary Computation 88 (2024) 101612

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo



Consensus-based virtual leader tracking swarm algorithm with GDRRT*-PSO for path-planning of multiple-UAVs

Berat Yildiz [a,*], Muhammet Fatih Aslan [a], Akif Durdu [b], Ahmet Kayabasi [a]

[a] Electrical Electronics Engineering, Karamanoglu Mehmetbey University, Karaman, Turkiye
[b] Robotics Automation Control Laboratory, Electrical Elektronics Engineering, Konya Technical University, Konya, Turkiye

ARTICLE INFO

Keywords:
Path-planning
Target detection
Collision avoidance
Formation Control

ABSTRACT

UAV technology is rapidly advancing and widely utilized, particularly in social and military domains, due to its extensive motion and maneuverability. Coordinating multiple UAVs enables more rapid and efficient task execution compared to a single UAV. The proliferation of UAVs across various sectors, including entertainment, transportation, delivery, and social domains, as well as military applications such as surveillance, tracking, and attack, has spurred research in swarm systems. In this study, a new swarm topology is presented by combining the Consensus-Based Virtual Leader Tracking Swarm Algorithm (CBVLTSA), which provides formation control in swarm systems, with the Goal Distance-based Rapidly-Exploring Random Tree with Particle Swarm Optimization (GDRRT*-PSO) route planning algorithm. Recently proposed, GDRRT* is notable for its efficient operation in expansive environments and rapid convergence to the goal. Within this framework, the path generated by GDRRT* is optimized using PSO to yield the shortest current route. CBVLTSA employs a potential push and pull function to facilitate cooperative, coordinated flight among swarm members. While applying pushing force to avoid collisions with each other and obstacles, members also exert pulling force to maintain flight formation while navigating to target points. This ensures controlled flight formation and collision-free traversal along the GDRRT*-PSO route. Consequently, unlike the others, the proposed algorithm achieves faster target reach with pre-planned routes, demonstrating a robust and flexible swarm topology with CBVLTSA. Moreover, we anticipate the significant utility of this algorithm across various swarm applications, including target detection, observation, tracking, trade and transportation logistics, and collective defense and attack strategies.

1. Introduction

Swarm robotics can enable complex applications to be performed faster and more efficiently, thanks to the multi-robot system working simultaneously [1]. When the coordination between the swarm members is ensured correctly and effectively, the failures caused by defective or faulty swarm members can be eliminated by those not connected to a single center [2]. In addition, using 3D airspace efficiently without causing air pollution is an essential advantage of using UAVs in swarm robotics. On the other hand, providing coordinated swarm mobility in large areas can be challenging. Therefore, an intelligent path-planning algorithm for the coordinated movement of the UAV swarm can offer highly effective results in commercial, military, and civilian areas.

Swarm robotics behaviors are similar to social behaviors, not in a specific order, and occur automatically [3]. Therefore, when swarm behavior is examined, it is expected to continue to work together against changes in the dynamic environment, even if some members are missing [4]. The main problem in the dynamic environment is updating the swarm parameters to suit the changes in the external environment [5]. The swarm system includes control parameters sensitive to swarm and environment size. Therefore, physical definitions such as behavior-based, virtual structure, leader following or consensus may be needed to control the swarm, especially in target-oriented tasks [6,7]. However, a significant limitation is that all vehicles used in target-oriented tasks have the same sensor footprint, and member loss is not considered..

Swarm systems are widely used in target detection, tracking and transportation applications. Researchers have generally preferred heuristic and probabilistic algorithms [8] in such applications. In a target detection application utilizing a Genetic Algorithm (GA), the swarm control parameters, generated from the dynamic and kinematic characteristics of members to achieve the maximum number of swarms

* Corresponding author.
E-mail addresses: beratyildiz@kmu.edu.tr (B. Yildiz), mfatihaslan@kmu.edu.tr (M.F. Aslan).

https://doi.org/10.1016/j.swevo.2024.101612
Received 20 December 2023; Received in revised form 29 April 2024; Accepted 21 May 2024
Available online 30 May 2024

reaching the target, are updated by minimizing the cost function extracted using Forward Euler time discretization method with GA [9]. The forward Euler method is a method used to find the numerical solution of a differential equation [10]. Although the solution converges to the actual value as the update step gets smaller in the Forward Euler solution, it is a crucial disadvantage that the computation time is prolonged. On the other hand, the update step can be chosen bigger to speed up the algorithm, but there may be an increase in system error as the number of updates decreases.

Organizing the inventory records of the stocks in the warehouses and determining their location is also a critical target detection problem. Time and cost savings can be achieved by using autonomous UAVs [11]. In studies where UAVs were used to organize warehouse inventory [12, 13], RFIDs could control indoor and outdoor areas. Moreover, Swarm UAVs can be a fast and low-cost tool for inventory control, package delivery, and transportation, as infrastructure and traffic do not affect them. Although swarm UAVs are sufficient for similar applications that will only be used for target detection [14], they are insufficient for swarm systems that require collective and coordinated action, including formation control and tracking.

The problems that has emerged in swarm robotics in recent years is that, unlike cooperative swarm members, Byzantine agents in a swarm can manipulate data and send wrong messages to neighbors, damaging the swarm topology. Shang [15] has proposed a control protocol that includes resilient tracking consensus, allowing collaborative agents to get closer to the Leader in the presence of these malicious Byzantine agents. The primary algorithm proposed in this work is the Weighted-Mean-Subsequence-Reduced (W-MSR) algorithm [15], which is designed to eliminate outliers in the neighboring space of any collaborative members. The authors also use a martingale convergence theorem to ensure that consensus is almost certainly followed in a fully distributed manner. However, in aerial swarm system applications, completing the sub-tasks before reaching the target becomes more critical, apart from the target-oriented tasks. Therefore, applications involving many sub-tasks, such as aggregation, formation control, and target research, should be planned from the beginning to the end of the swarm movements. In such applications, algorithms based on a Leader have been proposed, and the whole swarm dynamics are determined according to the Leader [16–20]. From this perspective, consensus-based algorithms for the formation and control of swarms have been proposed by Carli et al. [16]. This algorithm includes aggregation, formation control, trajectory tracking, and collision avoidance subtasks to cover the entire swarm movement [16]. However, in this study, no precaution was taken against the failure of the Leader or the inability of some of the members since the whole system's dynamics depend on the Leader. One of the main points that led to our study is this problem, which is ignored also in other studies [17–20]. In another application with subtasks, the Hierarchical Collaborative Fusion Navigation Algorithm (HCF), which can select a hierarchical leader instead of depending on a single leader, was preferred as a solution to this problem [17]. When the Leader disappears, a new leader is elected and all other members become his followers [17]. Although this ensures route safety during flock flight, changes in flight formation caused by fallen members may limit its application areas. This is one of the fundamental swarm problems called flexibility [9,21], which we tried to overcome in our study.

One of the primary purposes of swarming practices, such as target detection, transportation, etc., summarized in the previous paragraphs, is to discover the shortest path to reach the target or targets without colliding with the surrounding obstacles. From this perspective, the path planning problem concerning swarm systems arises. Researchers generally propose optimal or multi-modal path-planning algorithms in target or region-based path-planning applications [22]. Optimal path planning emerges as a prominent issue when dealing with singular static or dynamic targets [6], whereas multi-modal path planning captures attention when dealing with multiple static or dynamic targets [23]. A specific route is required for UAVs to perform a mission, which includes target detection or tracking, as a swarm. The optimum path between the initial and target points should be determined to generate motion information, considering the environment's obstacles and the space's structure [24]. In this context, various path-planning studies have been carried out for 2D and 3D environments for different mobile vehicles [25]. Evolutionary computation and swarm intelligence algorithms are generally preferred path-planning algorithms for swarm UAVs [26]. For example, they considered the paths to be followed by UAVs as polygons and combined GA and Fuzzy Logic (FL) to increase accuracy during path planning [27]. Here, swarm member UAVs are classified as Polygon Visiting Multiple Traveling Salesman Problem (PVMTSP).

There are studies where swarm intelligence algorithms and artificial intelligence are used together with a similar approach. For example, Liu et al., while creating the route with Artificial Neural Network (ANN), used GA to adjust the accuracy of the learning parameters of ANN [28]. There are studies to collaboratively identify targets for the swarm using mutation-first Differential Evolution (DE), which focuses on increasing genetic diversity compared to the GA algorithm [29]. On the other hand, it may become more challenging to find the most suitable route, and it may be more expensive. Another GA-based study [30] used the Immunity Genetic Algorithm (IGA) to determine the swarm route to detect dynamic targets during flights. However, due to system dynamics, sometimes the distance between divided roads can be very short. This causes sudden direction changes. Therefore, it can both reduce efficiency and increase the risk of collision. In this case, roads with sharp turns need to be smoothed again. The authors turned to heuristics swarm intelligence algorithms to smooth the paths and make the calculation speed efficient [31]. Cekmez et al. and Perez-Carabaza et al. used Ant Colony Optimization (ACO) to avoid complex obstacles when planning optimal UAV paths [26,32]. Li et al. propose particle swarm optimization (PSO) based on safe distance to ensure decentralized control of swarming UAVs in agricultural areas [33]. The path-planning techniques that stand out in these studies are node-based methods. Although the study's results were compelling, this was very costly and time-consuming on large maps. Recently, sampling-based methods have become more useful for applications that include UAVs and require 3D path planning, as they have less computational complexity, contain fewer nodes, and provide more optimal solutions on large maps. Therefore, in our study, the GDRRT* algorithm [31], a sampling-based path planning algorithm, is recommended because it has less computational complexity, contains fewer nodes, and provides more optimal solutions on large maps.

One of the sampling-based path planning methods to be best adapted to UAV applications is Random-Exploring Tree Star (RRT*) [34]. However, problems such as increased memory consumption in large spaces and slow convergence to the target have started the search for new alternatives to RRT* [35]. Despite RRT* versions such as Informed-RRT* [36], RRT*-Smart [37], PQ-RRT* [38], Cloud-RRT* [39], etc., Goal Distance-based RRT* (GDRRT*), which was recently developed by Aslan et al. [31], is remarkable for its fast convergence and ability in large spaces. Studies use RRT and improved RRT algorithms on swarm UAV applications [40–43]. Although alternatives to RRT* have been developed, each proposed new method has a shortcoming. Many algorithms increase the path planning time while shortening the path or are designed to work only in small and/or obstacle-free environments at the expense of shortening the time. Developing a fast path planning algorithm in environments containing large and complex obstacles is still challenging.

Recently, researchers have utilized optimization and artificial intelligence algorithms in existing path planning algorithms to optimize the path and provide faster planning. Wang et al. [44] presented the Neural RRT* algorithm. The method used the path plans produced by the A* algorithm in 2D environments as ground-truth data. Then, the environments and path plans produced by A* were fed into the CNN model. As a result, the authors proposed an algorithm that converges to the goal

faster. However, the shortcomings were that A* was cumbersome in large environments and that the work was limited to 2D environments. Flores-Caballero et al. [45] proposed a path planning study based on the Pruned A* algorithm on six different 3D environment scenarios. The generated paths were optimized with Differential Evolution (DE) and GA algorithms. However, this application was also limited to small environments; a* was very slow in large environments with obstacles. As mentioned before, Aslan et al. [31] not only proposed the GDRRT* method in their study, but also the authors optimized the path found by GDRRT* with PSO. As a result of the optimization, the applied GDRRT*-PSO method significantly reduced the planned path in large environments.

After successful path planning, each swarm member must interact with the members in its close neighborhood and provide data flow. It is suggested that such a communication topology will be achieved by consensus-based algorithms [46]. The swarm collects environmental data throughout its movement and interacts with members in its immediate neighborhood. The dataset obtained by this interaction can be determined using nonlinear optimization methods to determine the optimum route. Consensus-based studies offer solutions for various issues such as aggregation [18], formation [19,20], tracking [47], surveillance [48], and foraging [49]. In the literature, distributed containment control under state constraints in multi-agent networks is encountered before solving such problems. Different approaches are taken depending on the characteristics of the communication network used by the swarm or multiagent topology. In this context, Shang [50] addresses the problem using a gradient projection approach and designs distributed consensus protocols that implement state restriction and steer agents asymptotically to proportions in terms of pre-assigned scales. Here [50], the Lyapunov function is introduced to ensure scaled convergence and suitable conditions have been developed to ensure the equilibrium is inside the intersection of constraining sets. The method is then extended to multiagent systems with static reference values. Similarly, a study examining constraint problems for a weighted multiagent network consisting of continuous-time agents under situation constraints where informed and uninformed agents receive external inputs [51] proposed a new distributed gradient-based nonlinear controller set to ensure that each agent's state remains within the desired convex constraint set and achieves both constraint and consensus tracking. The author designs the control input for the multiagent system using matrix analysis, convex analysis, and Lyapunov theory [51]. However, in real-time applications, it is still an issue that needs to be developed due to the limitations in data flow and the disruptions that will occur in the system design. Therefore, an intelligent control system is required in path planning for swarm movement. In addition, the swarm should be robust enough to perform the desired task despite the failures caused by technical or external reasons and should be flexible enough to update its internal parameters against the dynamic changes that occur in the environment they are in [21]. In the study we suggested before [21], swarm members have a robust and flexible structure. However, in real-time applications in large areas, it has limited efficiency in terms of energy efficiency as the air residence time will be long. In addition, since the route planning is determined online during the flight, the time to reach the target may differ in each application with different routes other than the expected shortest route [21]. It is a relatively undesirable situation for UAVs that act in a coordinated manner as a swarm.

The characteristics of studies presented in the literature regarding swarm UAVs are provided in Table 1. Upon examining path planning studies for swarm UAVs, it is observed that the majority address the collision avoidance problem [9,16,18–21,31,40–43,47]. However, many studies often overlook collaborative, collective, and coordinated formation control [9,16,18–20,32,33,40–43]. Furthermore, many studies lack one or more of these qualities when considering robust swarm capabilities to sudden changes in swarm size and the flexibility ability to reform formations after such changes. Additionally, even in studies that meet some of these criteria, the use of limited maps is noticeable. However, our study can provide all these features with a CBVLTSA algorithm and GDRRT*-PSO.

Considering these shortcomings, the motivation we obtained led us to develop an intelligent control system for path planning in swarm movement within the scope of this study. To achieve this, we propose a sampling-based hybrid GDRRT*-PSO algorithm that enables faster path detection in vast areas and optimizes the detected path to its shortest form. Additionally, we present an enhanced version of the algorithm we previously proposed [21], named the Consensus-Based Virtual Leader Tracking Swarm Algorithm (CBVLTSA) algorithm, which facilitates coordinated, collective, and cooperative formation control along the tracked route to enable flight. From this perspective, the UAV swarm, distributed in the simulation environment, was initially organized into a specific formation using the aggregation algorithm. Then, the route determined by the GDRRT* algorithm was optimized using PSO. In this way, the shortest route to reach the target point was determined. CBVLTSA was used so that the swarm could control its formation flight without hitting obstacles while moving toward the target point. Additionally, simulations have proven that the proposed algorithms meet the robustness and flexibility criteria.

The contributions of the developed study can be summarized as follows:

- Swarm implementation of the GDRRT*-PSO method, which provides fast path planning in large environments.
- A swarm topology that is robust enough to perform the expected task despite the loss of members during swarm flight

Table 1
The features of the studies presented in the literature on swarm UAVs.

Approaches		Study	Map size	Robustness	Flexibility	Formation control	Collision avoidance
Physical	**Consensus-Based**	[16]	small	X	X	✓	✓
		[48]	small	X	✓	✓	✓
		[21]	small	✓	✓	✓	✓
	RRT	[40]	large	X	X	X	✓
		[41]	large	X	X	X	✓
	ImprovedRRT	[42]	large	X	X	X	✓
		[43]	large	X	X	X	✓
	GDRRT*-PSO	Our	large	✓	✓	✓	✓
Evolutionary Computation	**GA**	[9]	large	X	✓	X	✓
	GA+FL	[27]	large	X	X	X	X
	GA+ANN	[28]	large	X	X	X	✓
	DE	[29]	large	X	X	X	✓
	IGA	[30]	small	X	X	X	✓
Swarm Intelligence	**ACO**	[26]	large	X	X	X	X
	ACO	[32]	large	X	X	X	✓
	PSO	[33]	large	X	X	X	X

- A flexible swarm topology that can create new flight formations in response to a deteriorated flight formation for any reason.

In the second part of the publication, the methodology used is given. Simulation studies supporting the proposed algorithms were made in the third part, and the results were interpreted. In the last part, the conclusion part, the importance, and future aspects are emphasized.

2. Methodology

2.1. GDRRT* and GDRRT*-PSO path planning algorithm

This section briefly introduces the GDRRT* method used as a path planner in this study.

Although RRT* outperformed many methods in both time and computational cost, it suffered from both increased cost and path planning time as the size of the environment increased. The GDRRT* [31] is one of the many methods developed to address these shortcomings. The highlights of GDRRT* are that it is designed for UAV (i.e., compatible with 3D environments), fast convergence to the target, and low computational cost [31]

Unlike RRT*, GDRRT* has limited sampling randomness rather than utterly random sampling. In this way, it is aimed that the newly assigned sample converges to the target in a controlled manner. Controlled sample assignment is provided by a d value determined before the algorithm. Using d, the random sample assignment equation of RRT* is modified as in Eq. (1).

$$x_{rand} = \left(x_{current} - \frac{d}{2}\right) + rand(0,1) \times \left(\left(x_{current} + \frac{d}{2}\right) - \left(x_{current} - \frac{d}{2}\right)\right) \quad (1)$$

With this equation, unlike RRT*, the new sample is not randomly assigned across the entire space region. The new random sample (x_{rand}) can take a random value in a space of radius d/2 drawn centered on the current node ($x_{current}$). That is, the sample range is constrained. Thus, the random sample can take limited values in each axis of a 3D space (see Fig. 1).

Additionally, the nodes closest to the target point are continuously recorded to avoid unnecessary sampling. The random sampling calculation refers to the nearest node to the target. New nodes moving away from the target are not taken into account. In this way, each newly assigned node provides convergence to the target. In cases where there is an obstacle between the newly assigned sample and the target, the GDRRT* algorithm behaves like RRT*. For more detailed information, reference [31] can be used. The path created between the initial and target node as a result of GDRRT* converges to the target faster than RRT*. However, this path may be longer than the path produced as a result of RRT*. Because the nodes sampled with GDRRT* are also generated randomly. Therefore GDRRT* does not guarantee a shorter path. However, since GDRRT* only works on nodes sampled closer to the goal, it converges to the goal much faster than RRT*. It also uses much fewer nodes this way. As a matter of fact, in the study conducted by Aslan et al. [31], in tests conducted on many different 3D environments, the average path planning time with RRT* was 25.64 s, while this time was 8.69 s for GDRRT*.

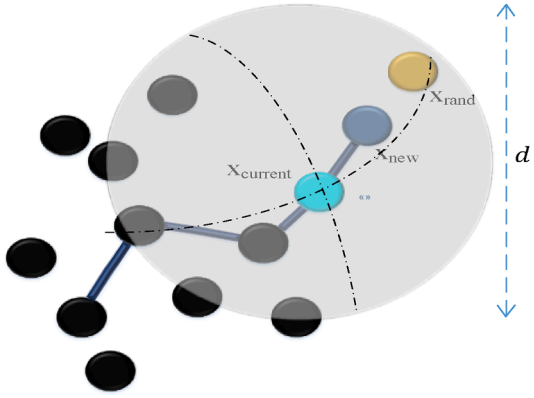


Fig. 1. GDRRT* algorithm [30].

Additionally, these values are supported by the average number of samples. Accordingly, while this value is 398.78 for RRT*, this value is 154.66 for GDRRT*. Considering that path length is important as well as planning time, the authors shortened the path planned with GDRRT* through PSO. Depending on the obstacle and space constraints, PSO successfully minimized the GDRRT* path. According to the authors, in tests conducted on many environments, the average path length produced by GDRRT* was 645.84, while for GDRRT*-PSO this value was 524.85. The PSO parameter values applied to GDRRT*-PSO in the original study are also used in our study with the same values. These parameters and their values are shown in Table 2.

Since possible environmental conditions may contain obstacles of different sizes in different places, in our previous study [31] the creation of the environment was carried out randomly based on the environment boundaries. Experiments for both GDRRT* and GDRRT*-PSO were conducted in these randomly generated environments. Additionally, the stability and robustness of both algorithms were tested with a total of 2000 3D random environment experiments. In these environments, the previously mentioned values of average path length, average plan time and average number of samples clearly show the superiority and differences of both GDRRT* and GDRRT*-PSO over RRT*.

As a result, the GDRRT* used in this application has made a great contribution to the RRT* algorithm, which is indispensable for 3D environments, in terms of speed. In addition to speed, PSO, which was applied to minimize the path length, shortened the path significantly. Fig. 2 can be examined to see the superiority of GDRRT* over RRT* and of GDRRT*-PSO over GDRRT*.

These benchmarks prove the validity of choosing the GDRRT*-PSO algorithm for swarm UAV applications where time and shortness of path are vitally important. Fig. 2(a) shows the path planning performances of GDRRT* and RRT* in the same 3D environment. GDRRT* has no clear advantage in terms of path length. However, for Fig. 2, the path lengths are 773.5 and 673.85 for RRT* and GDRRT*, respectively. The values for the number of samples and planning time that GDRRT* promises superiority are 50.38 s, 779, 15.39 s and 332 for RRT* and GDRRT*, respectively. For Fig. 2(b), the comparison between GDRRT* and GDRRT*-PSO can be made on path lengths for a different 3D environment. Accordingly, the path lengths are 591.84 and 528.08 for GDRRT* and GDRRT*-PSO, respectively.

2.2. Discrete-time consensus algorithm

Consensus-based swarm algorithm is an algorithm in which information flow is provided according to the neighborhood relations between the cooperative swarm members. In the case of $d_{ij} > 0$ where there is information flow from the i^{th} swarm member to its neighbor j, cooperative data flow updates are accepted as in Eq. (2) [52]. On the contrary, if $d_{ij} = 0$, it is assumed that there is no information.

Table 2
Parameter values of the PSO algorithm.

Control parameter	Value
Population size	150
Number of iterations	150
Acceleration coefficients (c1; c2)	1,5; 1,5
Inertia weight (w)	1

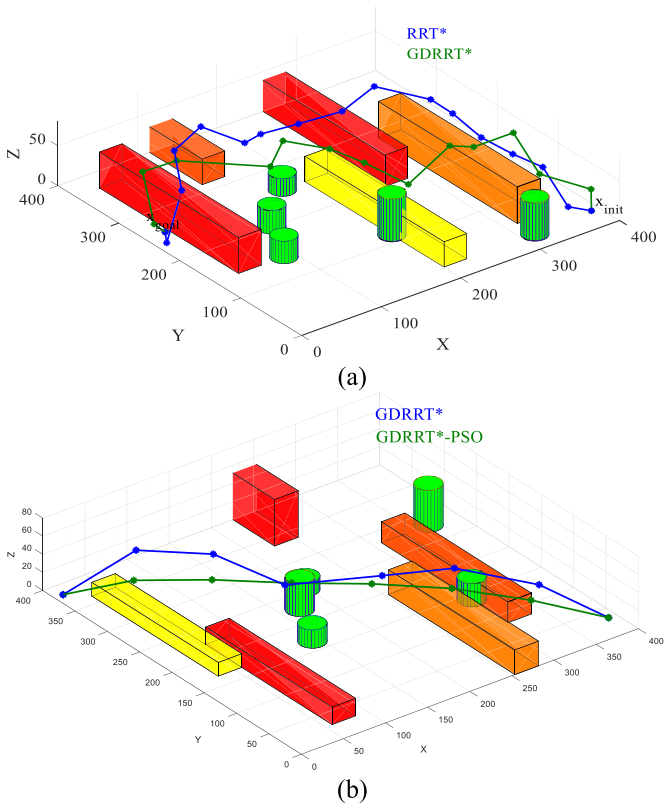


Fig. 2. Benchmarking of (a) RRT*&GDRRT* and (b) GDRRT*&GDRRT*-PSO.

$$C_i(t+1) = \sum_{j \in N_i} d_{ij} C_i(t) \tag{2}$$

When d_{ij} is transferred to a matrix to include the relationship between all neighboring swarm members, a row stochastic matrix D is formed with an eigenvector 1_m equal to 1 in all cases [53,54]. In this case, the weighted average of the current state of the member itself and the member in its neighbor N_i forms the updated consensus data $C_i(t+1)$ (3) [52].

$$C_i(t+1) = D\, C_i(t) \tag{3}$$

The matrix D has an eigenvalue of 1 when associated with the distributed directed graph structure G, and all remaining eigenvalues are inside the unit circle. $C(t)$ converges to $DC(0)$ when t diverges to infinity where D is equal to $1_m \mu^T$ and μ is the left unit eigenvector of D. Thus, it can be seen that each C_i value agrees on a common value given by $\sum_{i=1}^{m} \mu_i C_i(0)$ [54].

2.3. Consensus-based virtual lider tracking swarm topology

In UAV models, fundamental physical dynamics can be defined according to position, velocity, or acceleration. The discrete-time singular integrator dynamic model used in this application is given in Eq. (4) [52]. This provides convenience in terms of applicability and control of the communication topology.

$$x_i(t+1) = x_i(t) + u_i(t),\ i \in N,\ N = \{1,\ 2,\ 3,\ \ldots n\} \tag{4}$$

The instantaneous position of the i^{th} UAV in three-dimensional space is $x_i(t)$. The control input determines the following position $x_i(t+1)$ of the i^{th} UAV is $u_i(t)$. The size of the UAV swarm is expressed by N, a cluster that involves the swarm members, and each swarm member has an ε_r radius spherical sensor footprint.

In consensus-based swarm topologies, the holistic communication between members and the contribution of each member to the data flow on this topology provide significant advantages. A Distributed Directed Graph (DDG) is used in this topology where instantaneous location can be shared instantly within the swarm distribution [55]. Here, each member is in direct contact with other subset members. The basic graph structure is expressed as $G = (N,\ E)$, with the node set (N), which consists of swarm members, and edge set (E), which is the instant distance d between the members. The swarm, which is scattered, takes the form of a predetermined formation shape according to its size. Depending on the formation type, the distance between each i^{th} and j^{th} members d, is expressed as in Eq. (5) [54].

$$d_{ij} = ||x_i - x_j|| \tag{5}$$

In real-time, many robotic systems operating in swarms may encounter internal or external interventions. Therefore, it is imperative for the swarm to demonstrate robustness and flexibility [21] in such situations. We advocate the implementation of the Virtual Leader Tracking Algorithm (VLTA) to imbue these capabilities within the UAV swarm system [21]. With this algorithm, the swarm system can adeptly carry out its designated tasks by dynamically adjusting its parameters, even in the face of communication breakdowns or member malfunctions. The graph structure, founded upon the virtual leader, is denoted as $G_{VL} = (N,\ E) \cup \{V_L\}$ [21]. Fig. 3 illustrates the graph structures of the swarm topology established on virtual leader tracking.

In this context, the DDG structure is reconfigured based on the virtual leader's position to mitigate the computational density of geometric calculations. In the case of a three-UAV swarm, the virtual leader is positioned at the centroid of the triangular formation. Similarly, a virtual leader is placed in a four-UAV swarm at the formation's center. The arrangement of swarms comprising five or more members depends on whether the member count is odd or even. In the former case, the virtual leader is situated at the apex to establish a double-sided polygon, while in the latter case, it assumes a central position within the resulting double-sided polygon.

2.4. Aggregation and formation with potential push and pull functions

As the swarm moves toward the target, the formation shapes continuously adjust, considering obstacles and other swarm members to uphold optimal spacing. A target set $T_i(t)$ consisting of target points and an obstacle set $O_i(t)$ consisting of obstacle points are used to ensure formation control while moving to the target point. Each point, designated by the GDRRT*-PSO path planning algorithm, is denoted as target points $T_p(t)$. The cluster formed by these points is included in the $T_i(t)$ cluster expressed in Eq. (6).

$$T_i(t) = \bigcup_{p \in i} T_p(t), \quad p = \{1, 2, 3, \ldots \infty\} \tag{6}$$

UAVs were tried to be gathered at the same altitude so that the swarm members could reach location-based consensus among each other more quickly while reaching the target nodes one by one. While location updates are made, altitude control is also performed. This way, the swarm can be less affected by communication breakdowns caused by possible altitude loss. Additionally, state updates can be made more quickly via x-y coordinates throughout the simulation. Thus, the target points determined by the GDRRT*-PSO algorithm can be reached faster and safer. In order to collect the swarm at a fixed altitude, the z-axis

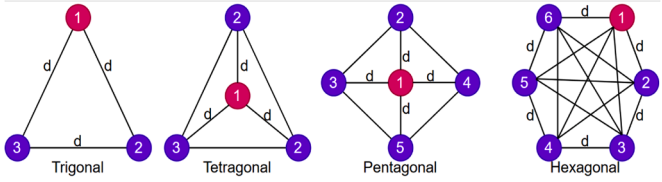


Fig. 3. Virtual leader-tracking based DDG [21].

coordinates of the nodes $z_p(t)$ on the route contained in the target set $T_p(t)$ are selected as the reference point. The average distance of each UAV to the z-axis coordinates of the nodes $z_{ap}(t)$ is calculated by Eq. (7) [21]. This value is subtracted from the desired coordinate point on the z-axis and determined as the target point of the virtual leader UAV with Eq. (8). When the position of the virtual leader UAV $z_i^*(t+1)$ converges to point $z_p(t)$, the $z_p(t+1)$ converges to 0. Thus, swarm members gather at a constant altitude.

$$z_{ap}(t) = \frac{\sum_{p=1}^{|z_p(t)|}\left[z_p(t) - z_j(t)\right]}{|z_p(t)|},\ p\ \epsilon T_p(t),\ j\epsilon N(t) \tag{7}$$

$$z_i^*(t+1) = z_p(t+1) = z_p(t) - z_{ap}(t),\ j\epsilon N(t),\ i = V_L(t) \tag{8}$$

Additionally, swarm members apply pushing and pulling forces to each other to avoid losing communication. If the distance between the members is larger than the desired formation distance, Eq. (9) is applied to determine the points $T_{ij}(t)$ that form the formation shape of the members [16].

$$T_{ij}(t) = \left\{x_j(t) + d_{ij}\frac{x_i(t) - x_j(t)}{||x_i(t) - x_j(t)||}\right\}, i\ \epsilon N(t),\ i \neq j \tag{9}$$

The formation points obtained with this equation are also included in the target set $T_i(t)$. Then, the pulling force given in Eq. (10) is applied so that the members can reach these points [16].

$$F_{ij}^T(x_i(t)) = \left\{\begin{array}{c} x_{1i}(t)x_{2i}(t) - x_{1i}(t)T_{2j}(t) + x_{1i}(t) - T_{1j}(t), \\ x_{2i}(t) - T_{2j}(t) \end{array}\right\}, \quad \text{i } \epsilon N(t)\ ve\ i \neq j \tag{10}$$

If the positions of the obstacles in the environment are known, they are added to the obstacle set $O_i(t)$ in Eq. (11). In addition, if the swarm members are closer to each other than the specified formation distance, the member positions are added to the obstacle cluster in Eq. (12) to avoid a collision, and the push force given in Eq. (13) is applied [16].

$$O_i(t) = \bigcup_{j\in N-\{i\}} O_{ij}(t) \tag{11}$$

$$O_{ij}(t) = x_j(t),\ i\ \epsilon N(t)\ ve\ i \neq j \tag{12}$$

$$F_{ij}^O(x_i(t)) = \left\{\begin{array}{c} x_{1i}(t)x_{2i}(t) - x_{1i}(t)O_{2j}(t) + x_{1i}(t) - O_{1j}(t) \\ x_{2i}(t) - O_{2j}(t) \end{array}\right\}, \quad i\ \epsilon N(t)\ ve\ i \neq j \tag{13}$$

2.4.1. Time-dependent adaptive-newton minimization

Position updates of the members during the swarm movement towards the target on the route determined by GSRRT*-PSO were provided using the time-dependent Newton minimization method. If member i has reached the destination point $T_{ij}(t)$ and communicating with neighboring members, it should be $x_i(t) = T_{ij}(t)$ and $O_{ij}(t) = x_j(t)$. In this case, the instantaneous push and pull functions $F_{ij}^T(x_i(t))$ and $F_{ij}^O(x_i(t))$ between swarm members should be 0. Thus, the equality $x_i(t+1) = x_i(t)$ is achieved and instant stable formation control is ensured. Within the scope of this application, the minimization formula applied to the control input u(t) is given in Eq. (14) [16].

$$x_i(t+1) = x_i(t) + \lambda\left(\Delta x_{pull}(t) + \Delta x_{push}(t)\right),\ i\ \epsilon\ N(t) \tag{14}$$

Here, λ is a constant step coefficient. $\Delta x_{push}(t)$ and $\Delta x_{pull}(t)$ form the instantaneous step vector of the potential push and pull functions, respectively, and are given in Eqs. (15)–(19) [16].

$$\Delta x_{push}(t) = \sum_{j\epsilon N-\{i\}} \overline{R}_{ij}(t) \tag{15}$$

$$\overline{R}_{ij}(t) = \left\{\begin{array}{c} \frac{R_{ij}(t)}{\left[1 + \left(\frac{||R_{ij}(t)||}{C_r}\right)^{\mu}.||R_{ij}(t)||^3\right]} - \frac{\varepsilon_r}{\left[\left(1 + \left(\frac{\varepsilon_r}{C_r}\right)^{\mu}\right).(\varepsilon_r)^3\right]},\ d_{ij} \leq \varepsilon_r \\ 0,\ d_{ij} > \varepsilon_r \end{array}\right\} \tag{16}$$

$$R_{ij}(t) = +\left[\nabla F_{ij}^O(x_i(t))\right]^{-1}.F_{ij}^O(x_i(t)) \tag{17}$$

$$\Delta x_{pull}(t) = \sum_{j\epsilon N-\{i\}} \frac{A_{ij}(t)}{||A_{ij}(t)||} \tag{18}$$

$$A_{ij}(t) = -\left[\nabla F_{ij}^T(x_i(t))\right]^{-1}.F_{ij}^T(x_i(t)) \tag{19}$$

C_r and μ given in Eq. (17) are constant push coefficients that can be determined according to the sensitivity of the environment. Moreover, the constant step coefficient λ can be defined adaptively $\lambda_i(t)$ in Eq. (20) from the instantaneous formation error $\varphi_i(t)$ difference $E_{fi}(t)$ in Eqs. (21) and (22) [16]. Thus, swarm location updates can be done more quickly.

$$\lambda_i(t) = \rho.e^{E_{fi}(t)}, \tag{20}$$

$$E_{fi}(t) = \varphi_i(t) - \varphi_i(t- -1), \tag{21}$$

$$\varphi_i(t) = \frac{\sum_{j\in N_i(t)}||d_{ij} - [x_i(t) - x_j(t)]||}{|N_i(t)|} \tag{22}$$

2.5. GDRRT*-PSO and virtual leader tracking algorithm

For target-oriented small or indoor applications, the real leader tracking method can produce successful results, where the entire communication topology is connected to the leader UAV [16]. On the other hand, to effectively process the common data of the swarm members in more extensive and vast areas, it is necessary to protect the flight formation by checking the alignment and reaching the target point [19,20]. Also, swarm members are susceptible to malfunctions or falls during real leader tracking as they move toward a specific target [21]. The formation shape may be distorted, or the communication topology may collapse. The slightest problem, especially in the leader UAV, can endanger the entire system. To avoid this situation, we propose a formation control algorithm based on a virtual leader like before [21]. The contribution made within the scope of this study is the creation of a hybrid algorithm by combining the CBVLTSA and the GDRRT*-PSO algorithm. In this way, the most optimum route for the swarm to use to reach the target point was determined. This is a significant improvement in terms of both energy and duty time.

With the proposed hybrid algorithm, swarm members continuously calculate the formation error after updating their positions relative to the virtual leader as they move to each GDRRT*-PSO node. In addition to the swarm members, the graph structure connected to the virtual leader, which is added depending on the formation shape, is constantly updated during the changes in the G_{VL} target and obstacle clusters. The potential pull force is rearranged to be $i = V_L(t)$ to follow the virtual leader as given in Eq. (23) [21].

$$F_j^T(V_L(t)) = \left\{\begin{array}{c} V_{1L}(t)V_{2L}(t) - V_{1L}(t)T_{2j}(t) + V_{1L}(t) - T_{1j}(t), \\ V_{2L}(t) - T_{2j}(t) \end{array}\right\}, \quad i = V_L(t)\ \Lambda\ i\ \epsilon\ N(t) \tag{23}$$

Taking into account the route determined according to the simultaneously optimized GDRRT*-PSO path planning algorithm, the virtual leader tracking algorithm provides the swarm movement. The flow chart of the proposed algorithms is presented in Fig. 4.

According to the flow chart, a graph structure is first created based on the location data of the members distributed in the environment

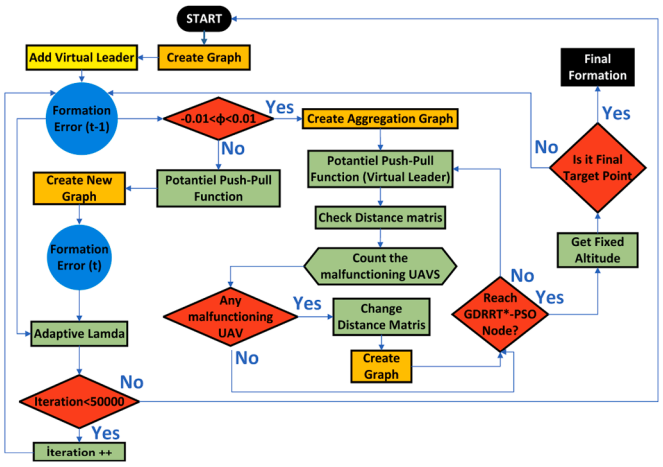


Fig. 4. Flow chart of consensus-based virtual leader tracking algorithm with GDRRT*-PSO.

(Create Graph). Afterward, a virtual leader is added to the graph structure (Add virtual leader). Potential push and pull forces are applied between swarm members toward the first node determined by the path planning algorithm. When the distance between swarm members reaches the desired formation distance, a flight formation graph that will be maintained throughout the flight is created (Create aggregation graph). After the swarm assumes the ideal flight formation, potential push and pull forces are applied between the virtual leader's position and the node points determined by the GDRRT*-PSO algorithm, allowing the virtual leader and follower drones to reach these node points.

Monitoring the distance between members throughout the swarm movement determines the number of malfunctioning or out-of-communication members (Count the malfunctioning UAVs). As the swarm reaches each node, it progresses to the next node. Thus, every point determined for the target is reached without colliding with obstacles, with the consensus-based virtual leader following the algorithm.

Assuming that some members lose communication at a random moment during the movement, the new formation creation function of the algorithm comes into play, and the swarm continues on its way by establishing a new graph structure and adopting a consensus-based formation once again. Therefore, the combination of the proposed CBVLTSA and GDRRT*-PSO algorithms facilitated rapid path planning on large-scale maps, ensuring successful navigation to the target point without encountering obstacles, while maintaining a robust and flexible swarm topology.

3. Simulations and results

Consensus-Based Virtual Leader Tracking and GGRRT*-PSO algorithms are combined within the scope of this application to present intelligent path planning for swarm UAVs. Through simulation tests, it has been demonstrated that the suggested technique enables more effective flights over bigger geographies. The algorithms suggested in this article are modeled in MATLAB. We adopt the following set of assumptions to facilitate applicability and observation:

(1) *UAV data was handled as points. The sensor radius, εr, is also considered as the safe distance.*
(2) *It is assumed during the simulations that the UAV, obstacle, and target point's locations are known in advance. The GPS and IMU sensors, which are frequently found in UAVs, can be utilized to determine the locations of the swarm member UAVs. The UAVs' infrared, ultrasonic, radar, or lidar sensors can locate obstacles depending on their present location.*
(3) *Each quadrotor must be connected to every other quadrotor on the network, which must be two-way.*

Since the environment map is large in the simulation, the communication sensor radius of the UAVs is assumed to be $\varepsilon r = 5$ and is randomly distributed in the environment. According to the position information obtained from each UAV, a formation was formed with a distance of $d = 5$ between them. The initial step coefficient (which is then updated depending on the adaptive error function) is taken as $\lambda = 0.1$, and the potential push function coefficients Cr and μ are taken as 1 and 0.1, respectively.

With the proposed algorithm, it is possible to detect or attack a specific target by following the leader in an environment with and without obstacles, as in other studies [16,21,52]. In these applications [16,21,52], the swarm gathers at a fixed altitude and forms the flight formation by following the real leader. Afterward, the real leader UAV is followed, and the predetermined target point is reached. Such studies can yield successful results against small and scattered obstacles [16, 21]. However, as the size of the map expands and the number of obstacles increases, it becomes more challenging to reach the target point. In addition, an inefficient situation may be encountered regarding energy, cost and flight time. For this reason, in previous studies [16,21,52], swarm flights that avoid simple obstacles on small-scale maps without path planning are inefficient in terms of energy and cost for large areas. Within the scope of this study, previously used algorithms [16,21] were developed and combined with intelligent path planning [31], and swarm UAV behaviors were tested against obstacles of different sizes on larger maps. In addition, thanks to the virtual leader used in the CBVLTA algorithm, even if there is a loss in the number of swarm members, the target point is reached by re-forming the flight formation with the remaining members.

3.1. Virtual leader tracking algorithm with GDRRT*-PSO

The simulation results based on the CBVLTSA algorithm with GDRRT*-PSO are presented in Figs. 5–7. The environmental mapping for the collaborative flight of the UAV swarm is depicted in Fig. 5a. Various obstacles of different heights and sizes are present in this environment. The position indicated by the red dot is the starting point where the UAV

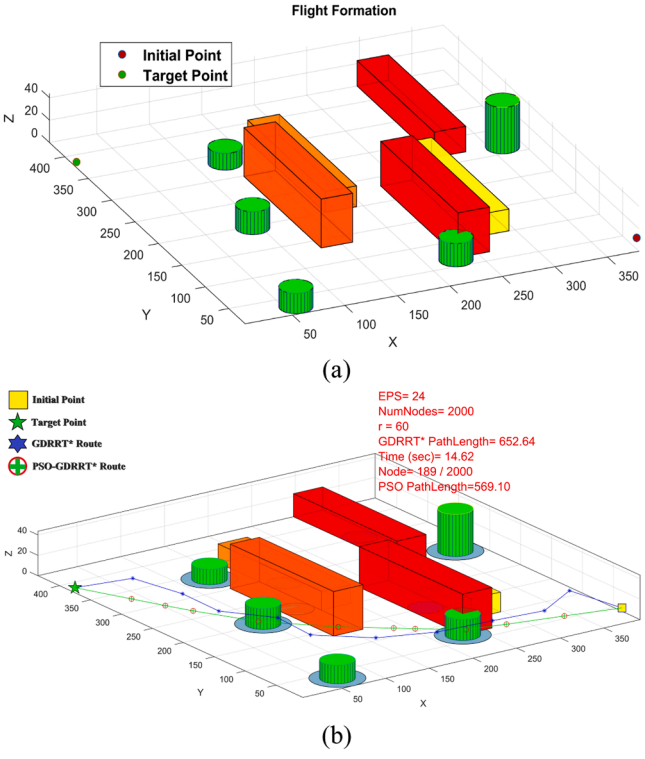


Fig. 5. GDRRT*-PSO route planning.

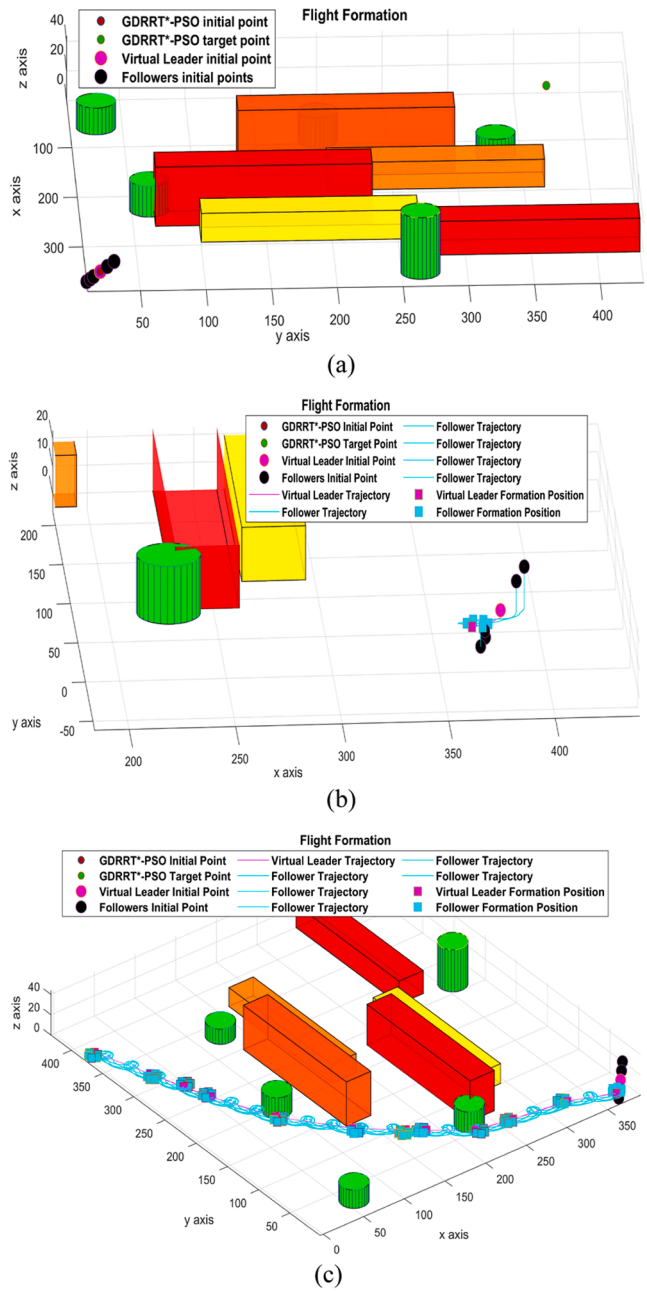


Fig. 6. Consensus-based virtual leader tracking and formation control in the 3D.

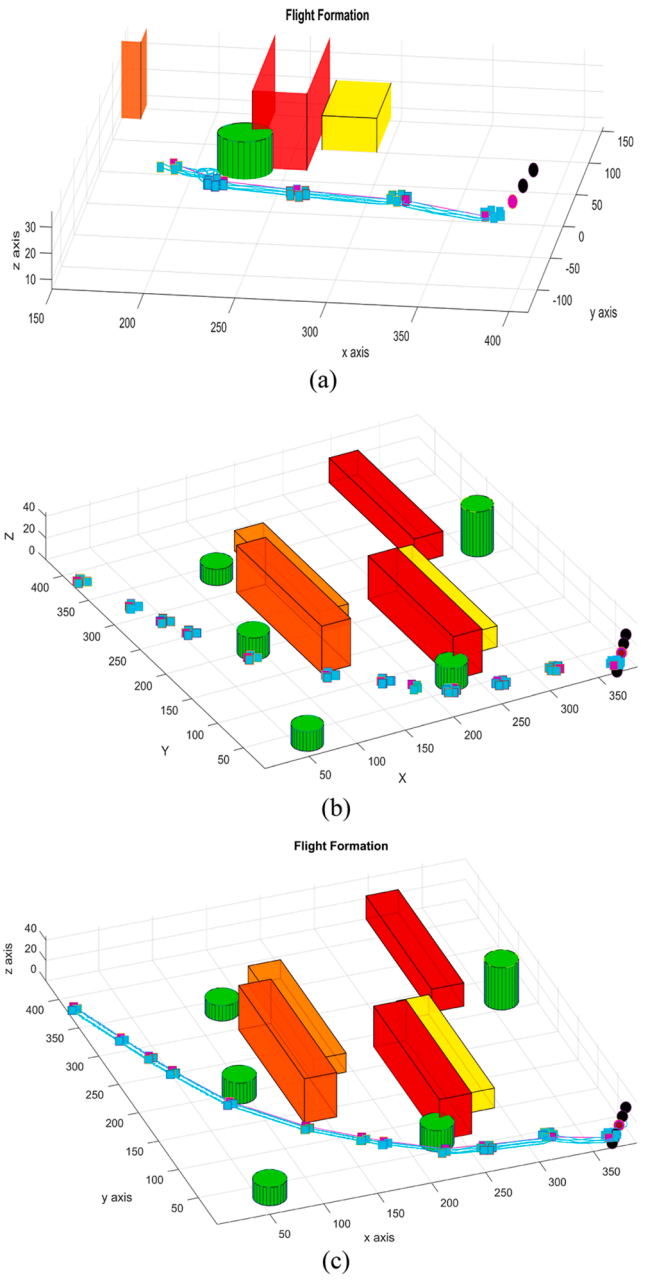


Fig. 7. Robustness and flexibility trial of virtual leader tracking and formation control algorithm.

swarm will gather and form a formation. The green dot represents the target point that the swarm needs to reach. The objective of the initial simulation is to maintain the flight formation among swarm members without colliding with obstacles and to reach the target point via the shortest path. To determine the shortest route, the GDRRT*-PSO algorithm is proposed. GDRRT*generates a route consisting of obstacle-free linearly connected points. However, this route is not the final route that swarm members will follow. Because the linear route between any two points on this route is taken randomly on unobstructed points, the overall route is longer. Subsequently, PSO is utilized to identify optimal short distances. Here, the number of nodes obtained with GDRRT* is assumed constant, and the shortest Euclidean distance between two nodes is optimized using PSO. Thus, a secure route is established for the swarm, ensuring that it reaches the target with the least energy expenditure in the shortest time. The route generated by the GDRRT*-PSO combination is presented in Fig. 5b.

In the initial stage, a swarm of 5 UAVs was randomly distributed (Fig. 6a). A virtual leader is assigned to the swarm of 5 UAVs before the aggregation algorithm based on potential push and pull functions is applied. Then, using the gathering and fixed altitude algorithm, the members form the hexagonal formation at the same altitude (Fig. 6b). With the proposed algorithms, the UAV swarm reached the target point quickly, maintaining its flight formation without hitting obstacles along the determined route (Fig. 6c).

To test the robustness and flexibility of the system, it is assumed that a random number of UAVs malfunction or lose communication at a random moment of flight time. This means that position data is not received from any UAV, and its value is 0 in the distributed directed graph structure, or a value ($d>>5$) is calculated far from the expected formation distance. Therefore, these are situations where consensus is not achieved. In the first scenario, where the communication of 2 UAVs was lost during the flight, thanks to the proposed CBVLTSA algorithm,

the swarm reached the target point without hitting the obstacle by assigning a virtual leader and forming a new tetragonal formation with the remaining 4 UAVs (Fig. 7a and b). Here, it can be said that the swarm is robust enough to complete the task despite losing members. It has also been observed that it has a flexible structure sufficient to reconstruct the deteriorated flight formation. In another similar scenario, it is seen that the communication of 3 UAVs is interrupted. Thanks to the proposed method, a virtual leader was assigned to the remaining 2 swarm members a trigonal flight formation was created and the flight was carried out to the target point (Fig. 7c).

3.2. Performance evaluation of swarm topology

Since there is holistic communication and coordination in swarm topologies, the time to reach the target point can be put forward as a performance measure in the first place. However, some critical situations may be encountered at this point. For simple target-oriented tasks, leader tracking can be done throughout the swarm movement without formation control [16]. But, in missions such as area scanning and collective attack or defense, formation control throughout the flight is more important [21]. In larger-scale applications, leader tracking, formation control and re-formation against member or communication losses that occur during the flight come to the fore. Considering the above-mentioned situations, a new topology based on intelligent path planning has been proposed with the algorithm developed for swarm UAVs. The algorithm proposed in this study addresses many areas where swarm UAVs can be used. For example, different formation shapes can be combined and used in many visual shows in the entertainment industry. Since it can successfully perform collective and coordinated movements, it can also be adapted to the logistics and delivery sector. In addition, since the proposed topology includes robustness and flexibility criteria, it can continue its mission by creating consensus among the remaining agents despite the agents lost during surveillance, tracking and attack applications in military areas. In short, in many areas where UAVs can be used in a swarm, they will be able to stand out with their fast route planning and robust and flexible communication topology.

In this study, the swarm performance criterion was evaluated according to the time to reach the target point, number of iterations, route length and swarm size. The evaluation table of simulation results is given in Table 3. First of all, UAVs randomly distributed in the environment move towards the first node suggested by the GDRRT*-PSO algorithm. During this movement, the proposed nodal point is determined as the location of the virtual leader and a hexagonal flight formation is created in line with the potential push and pull forces applied to the swarm members. The flight formations' observed periods to reach the target place are directly proportional to the size of the swarm. Time to target is reduced, particularly in simulations where UAVs are dropped to evaluate their robustness and adaptability. Additionally, it can be argued that the proposed path planning algorithm will result in a more advantageous time to reach the target point on a large map. Although in real leader-following practice [16], it is claimed that the target is reached faster by designing a simpler trajectory, this method is still susceptible to swarm system failures or communication interruptions. In addition, although the work we proposed before [21] is robust and flexible against swarm system failures or communication interruptions, it is disadvantageous regarding energy and cost due to coordinated collective flight on large maps as in this application.

As a result, the proposed algorithm provides a more robust structure against changes in the number of swarm members and is more flexible against changing formation flights compared to applications that follow real leaders [3,16,19,48]. In addition, compared to the previously proposed virtual leader-following path planning algorithm [21], it has a significant advantage in terms of energy and cost by providing faster path planning on larger maps thanks to the GDRRT*-PSO algorithm.

Table 3
The performance of swarm topology.

Swarm size		5 UAV	3 UAV dropped	2 UAV dropped
Flight formation	*Time (min)*	76	42	54
	Iteration	4100	2352	2916
	Virtual Leader Trajectory (unit)	570	570	570
	Follower 1 Trajectory	685	266	632
	Follower 2 Trajectory	711	284	654
	Follower 3 Trajectory	662	636	296
	Follower 4 Trajectory	706	638	644
	Follower 5 Trajectory	696	242	288

4. Conclusion

UAVs are technological tools in demand in many areas today, and it is crucial to deploy them in flocks. However, it can be challenging to use in a swarm due to its dynamic structure. In particular, malfunctions in swarm members and sudden changes due to external factors can cause significant problems. In this study, the flight formation control of swarmed UAVs is provided by CBVLTSA, while the flight route is created using GDRRT*-PSO, an intelligent path planning algorithm. A communication topology is introduced by giving consensus-based formation control during the swarm flights. In this topology, swarm members share their current location, which is read from their sensors. Throughout the movement, the swarm members try to maintain the formation by using the potential pushing and pulling force among themselves, while the swarm attempts to reach the target nodes in turn by using the potential pulling force between the target node and the virtual leader.

Integrating GDRRT*-PSO path planning into the extended CBVLTSA algorithm offered faster path planning on large maps compared to previous studies. This provides energy efficiency and low-cost advantages in terms of herd systems. Moreover, thanks to the CBVLTSA algorithm, implemented despite malfunctions or communication breakdowns in the swarm members, the formation shape was reconstructed during the simulation and controlled throughout the flight. Therefore, it can be said that due to simulation applications, intelligent path planning was made for swarm UAV systems, and a Robust and Flexible structure was created within the swarm topology. In this way, disruptions due to loss in the number of swarm members, which may be frequently encountered, especially in real-time applications, can be overcome. In addition, the flexible formation structure will provide convenience to the swarm for variable tasks.

In environments that are very complex in terms of obstacles, GDRRT* may move too close to obstacles to approach the target, increasing the chance of crashing and not being able to avoid the obstacle. Since GDRRT* focuses entirely on convergence towards the target, it may get stuck in space that is close to the target but contains obstacles. So it can be called a local minimum. Additionally, if the most important parameter of GDRRT*, the d value, is chosen too small, GDRRT* will converge to the target very slowly. Additionally, while GDRRT*-PSO significantly reduces the path length, additional time will be required for optimization.

In future studies, experiments on system performance can be conducted with real-time applications based on different swarm algorithms. More tasks can be performed simultaneously if the proposed CBVLTSA algorithm is developed and the main swarm is divided into sub-swarms for multiple target detection. Additionally, a comparison will be made with newly proposed swarm studies for the comparative analysis of the proposed CBVLTSA algorithm. In order to minimize time dependency in the path planning algorithm used, the use of artificial intelligence-based path planning algorithms is also among our future work plans.

CRediT authorship contribution statement

Berat Yildiz: Software, Methodology. **Muhammet Fatih Aslan:**

Software. **Akif Durdu:** Writing – review & editing. **Ahmet Kayabasi:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The authors thank RAC-LAB (www.rac-lab.com) for providing this study's trial version of their commercial software.

References

[1] Y. Mohan, S.G. Ponnambalam, An extensive review of research in swarm robotics, in: 2009 world congress on nature & biologically inspired computing (nabic), IEEE, 2009, pp. 140–145.
[2] L. Bayındır, A review of swarm robotics tasks, Neurocomputing 172 (2016) 292–321.
[3] M. Campion, P. Ranganathan, S. Faruque, UAV swarm communication and control architectures: a review, J. Unmanned Veh. Syst. 7 (2) (2018) 93–106.
[4] S. Jung, K.B. Ariyur, Robustness for large scale uav autonomous operations, in: 2011 IEEE International Systems Conference, IEEE, 2011, pp. 309–314.
[5] C. Sampedro, H. Bavle, J.L. Sanchez-Lopez, R.A.S. Fernández, A. Rodríguez-Ramos, M. Molina, P. Campoy, A flexible and dynamic mission planning architecture for uav swarm coordination, in: 2016 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2016, pp. 355–363.
[6] S.J. Chung, A.A. Paranjape, P. Dames, S. Shen, V. Kumar, A survey on aerial swarm robotics, IEEE Transact. Robot. 34 (4) (2018) 837–855.
[7] J. Zhang, X.I.N.G. Jiahao, Cooperative task assignment of multi-UAV system, Chin. J. Aeronaut. 33 (11) (2020) 2825–2827.
[8] H. Liu, Q. Chen, N. Pan, Y. Sun, Y. An, D. Pan, UAV stocktaking task-planning for industrial warehouses based on the improved hybrid differential evolution algorithm, IEEE Trans. Industr. Inform. 18 (1) (2021) 582–591.
[9] T.I. Zohdi, Multiple UAVs for mapping: a review of basic modeling, simulation, and applications, Annu Rev. Environ. Resour. 43 (2018) 523–543.
[10] D. Estep, The forward euler method, Pract. Analy. One Variab. (2002) 583–604.
[11] A. Gupta, T. Afrin, E. Scully, N. Yodo, Advances of UAVs toward future transportation: the state-of-the-art, challenges, and opportunities, Futur. Transport. 1 (2) (2021) 326–350.
[12] F. Guérin, F. Guinand, J.F. Brethé, H. Pelvillain, Towards an autonomous warehouse inventory scheme, in: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2016, pp. 1–8.
[13] T.M. Fernández-Caramés, O. Blanco-Novoa, I. Froiz-Míguez, P. Fraga-Lamas, Towards an autonomous industry 4.0 warehouse: a UAV and blockchain-based system for inventory and traceability applications in big data-driven supply chain management, Sensors 19 (10) (2019) 2394.
[14] W.Y.H. Adoni, S. Lorenz, J.S. Fareedh, R. Gloaguen, M. Bussmann, Investigation of autonomous multi-UAV systems for target detection in distributed environment: current developments and open challenges, Drones 7 (4) (2023) 263.
[15] Y. Shang, Resilient tracking consensus over dynamic random graphs: a linear system approach, Eur. J. Appl. Math. 34 (2) (2023) 408–423.
[16] R. Carli, G. Cavone, N. Epicoco, M. Di Ferdinando, P. Scarabaggio, M. Dotoli, Consensus-based algorithms for controlling swarms of unmanned aerial vehicles, in: International Conference on Ad-Hoc Networks and Wireless, Springer International Publishing, Cham, 2020, pp. 84–99.
[17] R. Wang, J. Du, Z. Xiong, X. Chen, J. Liu, Hierarchical collaborative navigation method for UAV swarm, J. Aerosp. Eng. 34 (1) (2021) 04020097.
[18] O. Soysal, E. Sahin, Probabilistic aggregation strategies in swarm robotic systems, in: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005, IEEE, 2005, pp. 325–332. SIS 2005.
[19] Y. Kuriki, T. Namerikawa, Consensus-based cooperative formation control with collision avoidance for a multi-UAV system, in: 2014 American Control Conference, IEEE, 2014, pp. 2077–2082.
[20] T. Nguyen, H. Nguyen, E. Debie, K. Kasmarik, M. Garratt, H. Abbass, Swarm Q-learning with knowledge sharing within environments for formation control, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.
[21] B. Yildiz, A. Durdu, A KAYABAŞI, Consensus-based virtual leader tracking algorithm for flight formation control of swarm UAVs, Turk. J. Electr. Eng. Comput. Sci. 32 (2) (2024) 251–267.
[22] X. Yao, W. Li, X. Pan, R. Wang, Multimodal multi-objective evolutionary algorithm for multiple path planning, Comput. Ind. Eng. 169 (2022) 108145.
[23] W. Li, T. Zhang, R. Wang, S. Huang, J. Liang, Multimodal multi-objective optimization: comparative study of the state-of-the-art, Swarm. Evol. Comput. 77 (2023) 101253.
[24] H. Sang, Y. You, X. Sun, Y. Zhou, F. Liu, The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations, Ocean Eng. 223 (2021) 108709.
[25] S. Lin, A. Liu, J. Wang, X. Kong, A review of path-planning approaches for multiple mobile robots, Machines 10 (9) (2022) 773.
[26] U. Cekmez, M. Ozsiginan, O.K. Sahingoz, Multi-UAV path planning with multi colony ant optimization, in: Intelligent Systems Design and Applications: 17th International Conference on Intelligent Systems Design and Applications (ISDA 2017) held in Delhi, India, Springer International Publishing, 2018, pp. 407–417. December 14-16, 2017.
[27] A. Sathyan, N.D. Ernest, K. Cohen, An efficient genetic fuzzy approach to UAV swarm routing, Unmann. Syst. 4 (02) (2016) 117–127.
[28] C. Liu, W. Xie, P. Zhang, Q. Guo, D. Ding, Multi-uavs cooperative coverage reconnaissance with neural network and genetic algorithm, in: Proceedings of the 2019 3rd High Performance Computing and Cluster Technologies Conference, 2019, pp. 81–86.
[29] M.G.C.A. Cimino, A. Lazzeri, G. Vaglini, Using differential evolution to improve pheromone-based coordination of swarms of drones for collaborative target detection, in: Proceedings of The 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2016), Scitepress, 2016, pp. 605–610.
[30] Z. Zhou, D. Luo, J. Shao, Y. Xu, Y. You, Immune genetic algorithm based multi-UAV cooperative target search with event-triggered mechanism, Phys. Commun. 41 (2020) 101103.
[31] M.F. Aslan, A. Durdu, K Sabanci, Goal distance-based UAV path planning approach, path optimization and learning-based path estimation: GDRRT*, PSO-GDRRT* and BiLSTM-PSO-GDRRT, Appl. Soft. Comput. 137 (2023) 110156.
[32] S. Perez-Carabaza, E. Besada-Portas, J.A. Lopez-Orozco, J.M. de la Cruz, Ant colony optimization for multi-UAV minimum time search in uncertain domains, Appl. Soft. Comput. 62 (2018) 789–806.
[33] X. Li, Y. Zhao, J. Zhang, Y. Dong, A hybrid PSO algorithm based flight path optimization for multiple agricultural UAVs, in: 2016 IEEE 28th international conference on tools with artificial intelligence (ICTAI), IEEE, 2016, pp. 691–697.
[34] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Rob. Res. 30 (7) (2011) 846–894.
[35] J. Fan, X. Chen, X. Liang, UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased, Expert. Syst. Appl. 213 (2023) 119137.
[36] J.D. Gammell, S.S. Srinivasa, T.D. Barfoot, Informed RRT: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic, in: 2014 IEEE/RSJ international conference on intelligent robots and systems, IEEE, 2014, pp. 2997–3004.
[37] F. Islam, J. Nasir, U. Malik, Y. Ayaz, O. Hasan, Rrt*-smart: rapid convergence implementation of rrt* towards optimal solution, in: 2012 IEEE international conference on mechatronics and automation, IEEE, 2012, pp. 1651–1656.
[38] Y. Li, W. Wei, Y. Gao, D. Wang, Z. Fan, PQ-RRT*: an improved path planning algorithm for mobile robots, Expert. Syst. Appl. 152 (2020) 113425.
[39] D. Kim, J. Lee, S.E. Yoon, Cloud RRT*: sampling cloud based RRT, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 2519–2526.
[40] M. Kothari, I. Postlethwaite, D.W. Gu, Multi-UAV path planning in obstacle rich environments using rapidly-exploring random trees, in: Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, IEEE, 2009, pp. 3069–3074.
[41] A. Kaur, M.S. Prasad, Path planning of multiple unmanned aerial vehicles based on RRT algorithm, in: Advances in Interdisciplinary Engineering: Select Proceedings of FLAME 2018, Springer, Singapore, 2019, pp. 725–732.
[42] X. Ren, L. Tan, S. Jiaqi, X. Lian, Multi-target UAV path planning based on improved RRT algorithm, J. Phys.: Conferen. Ser. 1786 (1) (2021) 012038. IOP Publishing.
[43] W. Zu, G. Fan, Y. Gao, Y. Ma, H. Zhang, H. Zeng, Multi-uavs cooperative path planning method based on improved rrt algorithm, in: 2018 IEEE international conference on mechatronics and automation (ICMA), IEEE, 2018, pp. 1563–1567.
[44] J. Wang, W. Chi, C. Li, C. Wang, M.Q.H. Meng, Neural RRT*: learning-Based Optimal Path Planning, IEEE Transact. Autom. Sci. Eng. 17 (4) (2020) 1748–1758, https://doi.org/10.1109/TASE.2020.2976560. Oct.
[45] G. Flores-Caballero, A. Rodríguez-Molina, M. Aldape-Pérez, M.G. Villarreal-Cervantes, Optimized path-planning in continuous spaces for unmanned aerial vehicles using meta-heuristics, IEEE Access. 8 (2020) 176774–176788, https://doi.org/10.1109/ACCESS.2020.3026666.
[46] D.T. Davis, T.H. Chung, M.R. Clement, M.A. Day, Consensus-based data sharing for large-scale aerial swarm coordination in lossy communications environments, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2016, pp. 3801–3808.
[47] M.A. Ma'Sum, G. Jati, M.K. Arrofi, A. Wibowo, P. Mursanto, W. Jatmiko, Autonomous quadcopter swarm robots for object localization and tracking, in: MHS2013, IEEE, 2013, pp. 1–6.
[48] Petit, J. (2016). Distributed consensus-based formation control of quadrotors with formation feedback. nov. de.
[49] Q. Lu, J.P. Hecker, M.E. Moses, Multiple-place swarm foraging with dynamic depots, Auton Rob. 42 (4) (2018) 909–926.
[50] Y. Shang, Scaled consensus and reference tracking in multiagent networks with constraints, IEEE Trans. Netw. Sci. Eng. 9 (3) (2022) 1620–1629.
[51] Y. Shang, Consensus tracking and containment in multiagent networks with state constraints, IEEE Transact. Syst. Man Cybernet.: Syst. 53 (3) (2022) 1656–1665.

[52] T. Yang, Y. Wan, H. Wang, Z. Lin, Global optimal consensus for discrete-time multi-agent systems with bounded controls, Automatica 97 (2018) 182–185.
[53] F.F. Lizzio, E. Capello, G. Guglieri, A Review of Consensus-based Multi-agent UAV Implementations, J. Intell. Robot. Syst. 106 (2) (2022) 43.
[54] F.F. Lizzio, E. Capello, G. Guglieri, A review of consensus-based multi-agent uav applications, in: 2021 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2021, pp. 1548–1557.
[55] Z. Li, G. Wen, Z. Duan, W. Ren, Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs, IEEE Trans. Automat. Contr. 60 (4) (2014) 1152–1157.