A Particle Swarm Optimization-Based Cooperation Method for Multiple-Target Search by Swarm UAVs in Unknown Environments

2021 7th International Conference on Automation, Robotics and Applications (ICARA) | 978-1-6654-0469-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICARA51699.2021.9376529

Aniket Gupta
Delhi Technological University
New Delhi, India
aniket.tcdav@gmail.com

Aman Virmani
Delhi Technological University
New Delhi, India
virmaniaman4@yahoo.com

Parth Mahajan
Delhi Technological University
New Delhi, India
mahajanparth0@gmail.com

Raghava Nallanthigal
Delhi Technological University
New Delhi, India
nsraghava@dtu.ac.in

***Abstract*—This paper presents a decentralized method for multi-target search problem using a swarm of unmanned aerial vehicles with the information available from the onboard sensors. The proposed method deals with three main objectives: Time optimized multi-target search, optimized payload drops and inter-UAV collision avoidance which is independent of the number of UAVs. The proposed controller uses a modified Particle Swarm Optimization for cooperative multi-target search and is called Multi-Target Particle Swarm Optimization (MTPSO). The controller's performance is evaluated on Ardupilot's SITL platform for realistic simulations in various case-scenarios. Moreover, the performance of the controller with various inertial functions is also analyzed for different case-scenarios. Finally, the effectiveness of the proposed algorithm is illustrated by comparison with existing search methods. 0**

Keywords—Swarm, Unmanned Aerial Vehicles, Multi-target Search, Multi-agent systems, Particle Swarm Optimization

NOMENCLATURE

UAV	Unmanned Aerial Vehicle
PSO	Particle Swarm Optimization
NED	North East Down
FOV	Field of View
FFOV	Frontal Field of View
v_i	Velocity of i^{th} UAV
p_i	Position of i^{th} UAV
W_{start}	Search Area Starting waypoint
W_{end}	Search Area Ending waypoint
D	Distance between two UAVs
D_C	Communication range of UAV
D_{end}	Distance between UAV and end waypoint
D_{wap}	Waypoint completion radius
D_{safe}	Minimum allowable inter-UAV distance
D_{max}	Maximum displacement range
R	Random Coefficient
C	Acceleration Coefficient
O_{min}	Minimum overlapping constant
N	Total number of UAVs
N_S	Sub-swarm size
α	Angle of FFOV
v_{max}	Maximum velocity for UAV
v_{min}	Maximum velocity for UAV

I. INTRODUCTION

Unmanned Aerial Vehicles are increasingly becoming popular for their contribution in civil and military applications. Rapid area coverage, better surveillance and efficient monitoring are some of the characteristics that gives UAVs and edge over other alternatives. Ground-target search problem is arguably one of the most important applications of UAVs. This problem becomes increasingly complex as the number of UAVs and targets increase. In scenarios such as target-following and search rescue missions, it is important to provide rapid response with the ability to detect multiple targets.

Traditional methods [1] for detecting multiple targets by swarm of UAVs include simple exhaustive search wherein UAVs simply scan the complete search area using pre-defined trajectories. Though successfully deployed for target search, this method lacks in several ways. First, since the trajectories are predefined, they are not optimal and thus fail to respond in the shortest time for both surveillance and payload drop. Second, since the UAVs are not making any decisions themselves, they always require multiple operators to coordinate the complete mission which is both time-consuming and expensive.

Other approaches include dividing the complete search area into small cells and associate each cell with a probability of existence of target in the cell. Thus, creating a complete probability map for the complete search area. As the UAVs continue to progress in the environment, the probability of cells with targets are updated continuously, the cells with the highest probability are finally accepted as the detected targets. This method requires a centralized probability map which stores data from all agents and thus requires a robust fully connected network [2][3].

In [4], a similar approach is presented for groups of UAVs with limited sensing and communication capabilities. The probability associated with each cell is modelled using Bernoulli distribution. Each agent keeps its individual probability map generated by fusing data from other agents in

its communication range. Though effective, this convergence of agents on the targets is slow and the method itself is computationally expensive. Thus, it is not suitable for modern UAVs which focus on minimalistic architecture and light hardware.

Particle Swarm Optimization algorithms have been proven to provide best search performance but its usage in multi-target search has been limited due to lack of proper structure, multi-UAV collision and poor navigation in the unknown search area. Through this paper, we aim to tackle these problems. The proposed controller follows a defined structure while incorporating smooth inter-UAV collision avoidance, flocking and multi-target search with optimized payload drop. The algorithm is scalable, fault-tolerant and computationally light- weight and thus can be deployed on very light hardware components which is often the key feature of a UAV swarm.

Inertial functions used in PSO algorithm control the exploration and exploitation characteristics of the algorithm and thus using different functions results in varying performances [5]-[11]. The paper also proposes different inertial functions which can be used in various different scenarios in the surveillance region such as when the difference between the number of targets and the agents is very high or if an immediate response is required for the targets.

The rest of the paper is organized as follows: Section II gives a short introduction to the original Particle Swarm Optimization algorithm and leads on for the mathematical formulation of our problem statement. Section III presents the structure and design of the cooperative navigational controller for a single UAV. The Simulation in the loop (SITL) experimental results are presented in Section IV. Section V provides a brief conclusion, with recommendations for future work.

II. PRELIMINARIES

A. Particle Swarm Optimization Algorithm

Particle Swarm Optimization is a well-established optimization algorithm that iteratively tries to find the global optimum. The original algorithm uses a population of particles which move around the search space according to the PSO update rules. Each particle's movement is affected by its best-known position and the global best-known position. These values also get updated as better solutions are found by the particles. It is a metaheuristic algorithm and thus does not guarantee convergence to an optimal solution.

The PSO update rule is as follows:

$$v_i^{t+1} = \omega * v_i^t + R_1 * C_1 * (P_{best}^t - p^t) + R_2 * C_2 * (G_{best}^t - p^t) \quad (1)$$

B. Problem Formulation

We consider that a swarm of quadcopter UAVs have to navigate through a search area characterized by a starting waypoint W_{start}, ending waypoint W_{end} and the breadth B, searching for multiple targets scattered throughout the region. All the UAVs are assumed to be on the same height, thus moving in the same plane. The position vector for each agent can be denoted by $p_i^t = [x_i, y_i]^T$ and the velocity vector by $v_i^t = [\dot{x}_i, \dot{y}_i]^T$, for the UAV i (i = 1, 2, …, N) at time t, where N is the number of UAVs.

Each UAV is equipped with an onboard camera that can detect targets within its FOV and localize them using an onboard software stack*. The UAVs are also equipped with a single payload each and can drop it at any one target location. Thus, the complete mission UAVs is to scan the surveillance region, detect the maximum number of targets, drop maximum payloads and reach the end of the search area while avoiding inter-UAV collisions.

The network topology of all the agents is modelled by a partially connected ad-hoc network [12]. Mathematically the model can be represented by an undirected graph $G = (V, E), where\ V = \{1,2, \dots, N\}$ is the set of nodes and $E = \{(i,j): i,j \in V,\ ||p_i - p_j|| \leq D_C\}$ is the edge set. A single UAV directly connects only to its neighbouring agents denoted by $N_G = \{j \in V, (i,j) \in E\}$.

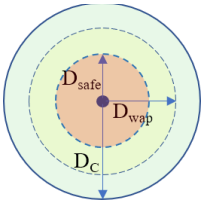

Fig. 1. Pictorial representation of D_{safe}, D_{wap} and D_C

Agents only share minimal data between themselves which includes their position data for collision avoidance, the position of the detected targets for cooperative convergence on target clusters and their payload state (see Table I) for cooperative payload delivery.

The algorithm is constrained by the following bounding conditions (in order of priority):

1. Avoid inter-UAV collision
2. Detect all the targets in the search area.
3. Drop the maximum number of payloads as soon as possible. (Rapid response)
4. Reach the end of the search area, so that no UAV wanders off.

III. ALGORITHM DESIGN

A. Initialization of UAVS

To maximize the exploration, we start the target-search process in an exhaustive fashion, i.e. UAVs are arranged in a straight line at the start of the search area. The overlapping between the FOV of each UAV is kept at a minimum overlapping constant (Omin). If the search area is much larger, it can be divided into multiple parts for multiple consecutive runs.

B. Inertial Term

The Inertial velocity term as the name suggests helps maintain the original course of the swarm unit and prevents rapid changes in the trajectory of each UAV. It is the product

of current velocity and the inertial function ω. Inertial function controls the exploration vs exploitation characteristics of the algorithm and is generally tuned to provide better exploration in the starting and exploitation as the swarm units become aware of the environment.

$$v_I^{t+1} = \omega * v^t \tag{2}$$

C. Personal Best

The personal best term is the best observation recorded by the swarm unit itself, it allows the units to explore the area under their current trajectory. While in traditional PSO, personal best is chosen out of all the observations, our algorithm follows a different approach. We only select the personal best from the current batch of observations, it provides the following two advantages:

1. It reduces the magnitude of the velocity vector offered by the personal best term and thus allows a stable trajectory for the UAVs.
2. It allows detecting multiple targets in the environment. Since each UAV follows the detect and drop method, the targets detected prior to the current observation can have one of the following three conditions only:
   a. UAV has dropped the payload to the target
   b. Neighbor UAV is coming to drop the payload to that target
   c. UAVs are out of payload and the target position is identified for further missions.

In each of these cases, the best strategy for multi-target detection is to choose the personal best from the current observations.

$$v_P^{t+1} = R_1 * C_1 * (P_{best}^t - p^t) \tag{3}$$

D. Global Best

To detect multiple targets in the environment all the swarm units are autonomously divided into sub-swarms of maximum size NS, which is a manually tunable parameter. UAVs are grouped into sub-swarms based on the inter-UAV distance if it is under Dmax.

The global best term represents the best observation detected by all of the UAVs in the sub-swarm. Similar to the personal best, MTPSO selects the global best term out of the current observations of the UAVs only.

$$v_P^{t+1} = R_2 * C_2 * (G_{best}^t - p^t) \tag{4}$$

E. Payload Optimization

Payload drop is the most important task in the rapid response strategy. For task assignment and optimized delivery, we start with assigning the following three states for the UAVs which helps in identifying which UAVs should continue search and which should go for payload drop:

TABLE I. PAYLOAD STATE

State	Meaning
State '1'	Payload available
State '0'	Going to the target position to drop the payload.
State '-1'	Payload unavailable

The position of the detected targets is continuously shared between the UAVs and the payload delivery task is assigned to the UAV nearest to the target within the constraint that target lies within the max displacement range.

$$v_{drop}^{t+1} = C_3 * (p_{drop} - p^t) \tag{5}$$

Further, if the UAV while going to drop payload on a target A, detects a target B which is closer than target A, it changes its drop location to target B, and target B becomes open for payload drop again. During payload drop, UAV continues to search for targets and sharing data with other UAVs. After dropping the payload, it does not retrace its path instead continues the search from drop point only.

F. Inter-UAV Collision Avoidance

Inter-UAV collision avoidance is necessarily required in any swarm system to ensure their safety. Thus, we utilize the concept of consensus equation to create an additional velocity term which can be vectorially added to the final velocity of each agent [13]. Avoidance is triggered if the distance between any nodes becomes smaller than Dsafe. This approach is easily integrable in our framework and can be represented as follows:

$$D = \left|\left|p_i - p^t\right|\right|_2$$

$$v_C^{t+1} = \begin{cases} C_4 * \sum_{i=1}^{N} (p_i - p^t) * \left(1 - \frac{D_{safe}}{D}\right), D \le D_{safe} \\ 0, D > D_{safe} \end{cases} \tag{6}$$

This approach only takes in account the agents approaching for collision and thus is suitable for our application. The Dsafe parameter is generally kept commensurate to the maximum swarm velocity and can be manually tuned if required.

G. Global Best and Personal Best Optimization

In the original PSO algorithm, Personal Best and Global best refers to the best personal value obtained and the best global value obtained over the entire iteration. This helps the particles to converge to a single optimum. In our case, we have modified this approach to dynamically update the Personal best and Global best.

We define a Frontal Field of View (FFOV) of each UAV with a front span of (-α, α). Each UAV constantly calculates the bearing of Personal best and Global best term wrt. To its position and if the bearing lies out of the frontal FOV range, both the terms are updated again. This is done to prevent the UAVs from settling on the pre-detected targets and continue the search for new targets. See Figure 2.

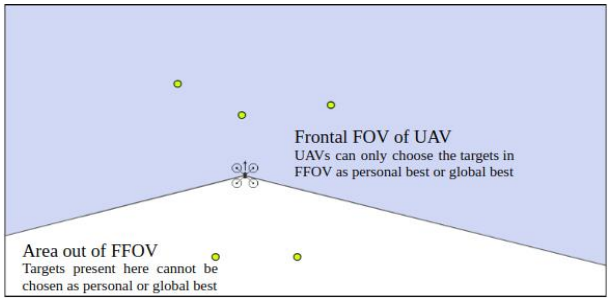

Fig. 2. Pictorial representation of FFOV

H. Endpoint Velocity

The UAVs need to reach their desired end waypoint in order to successfully complete the mission, therefore, they need a heuristic term that can guide them towards the goal waypoint even when they are detecting the targets. To achieve this, we introduce another vectoral velocity term given by the following equation:

$$D_{end} = \left|\left|p_{end} - p^t\right|\right|_2$$

$$v_{end}^{t+1} = C_5 * (p_{end} - p^t) * \left(1 - \frac{D_{wap}}{D_{end}}\right) \quad (7)$$

Algorithm 1 Multi Target Search PSO

```
Initialize the swarm unit
Initialize hyper-parameters
while distance_from_end_waypoint > waypoint_radius do
    Collect data from neighbours
    Record the sensor readings
    Concatenate targets detected by all UAV's in target list
    Filter the target list for duplicates
    Remove all targets present outside the frontal FOV
    for target from Detected_targets_list do
        Find M nearest UAV's
        Update Personal best
        Update Global best
        Assign UAV's to the nearest detected targets
    end for
    if UAV has to drop payload then
        if Payload available then
            Calculate velocity vector to go to Drop Location
            if Distance from drop location < Drop_distance then
                Drop payload
            end if
        end if
    else
        Calculate Final velocity
    end if
    Limit Final velocity
end while
```

I. Net Equivalent Velocity

The vectoral sum of Eq. 2 to Eq.9 can be calculated to find the final velocity of each UAV *i* as follows:

$$v_{i,final}^{t+1} = \begin{cases} v_I^{t+1} + v_P^{t+1} + v_G^{t+1} + v_C^{t+1} + v_{end}^{t+1}, state \neq 0 \\ v_{drop}^{t+1} + v_C^{t+1} \qquad\qquad\qquad\quad , state = 0 \end{cases} \quad (8)$$

On top of this superposition, we introduce the maximum and minimum velocity cut-offs which keep the direction of the desired vector but limit its magnitude between the desired values. The final equation thus can be represented as:

$$v = \frac{v}{|v|} * \min(max(v_{min}, |v|), v_{max}) \quad (9)$$

IV. Simulation and Testing

In this section, we first describe our method of simulations. Then, we quantitatively compare the effects of different inertial functions with varying FOV. Lastly, we demonstrate the effectiveness of MTPSO by comparing them with other methods in various simulated scenarios.

A. Simulation Environment

We use Ardupilot's Simulation in the Loop (SITL) software for all the simulations. The targets were generated using a random function in the defined search area. These targets were then plotted on the SITL map module by using a separate custom made Mavlink module. Each UAV while in the search area, scans the area around it using a search function in the range of its Field of View. The SITL simulator uses GPS coordinates for localization and thus a conversion from GPS to UTM coordinates was performed for linearization of calculations. See Figure 4 in Appendix.

B. Quantitative Comparison with Various Hyperparameters

Since inertial functions can drastically alter the performance of the algorithm, 5 inertial functions were carefully chosen to test their convergence properties, advantages and disadvantages. These experiments were coupled with varying the FOV of the camera module. FOV values taken for the experiment were chosen by keeping in mind the minimum overlapping criteria (O_{min}).

Two types of objectives were kept in mind for the experiments. First, to identify the best inertial function which can provide a good balance between exploration and exploitation when the number of targets equals the number of UAVs. Second, to observe the performance of the algorithm with increasing FOV.

The results of experiments are summarized in Table II. data recorded clearly shows that increasing the FOV increases the exploration as most of the search area is completely scanned, this provides better results in the total number of detected targets but also increases the total time taken to completely scan the environment.

C. Quantitative Comparison with Other Approaches

We compare our method with the basic Exhaustive Search Approach and Probability Graph approach on various scenarios. Table III summarizes the values of hyper-parameters used in simulation tests for MTPSO.

TABLE II. Performance Metrics Evaluated for Different Inertial Functions with Different FOV's

Metrics	Method	FOV (in meters)			
		55x32	70x40	91x52	140x80
Targets Detected	Constant*	19	20	20	20
	Random Inertial*	19	20	20	20
	Squared Decreasing*	19	20	20	20
	Sigmoid Increasing*	20	19	20	20
	Chaotic Inertia Weight*	19	20	20	20
Payload Dropped	Constant	18	19	18	20
	Random Inertial	18	17	18	19
	Squared Decreasing	19	17	18	20
	Sigmoid Increasing	17	17	18	18
	Chaotic Inertia Weight	18	17	18	20
Time for mission	Constant	209	206	205	208
	Random Inertial	213	208	211	212
	Squared Decreasing	213	209	211	215
	Sigmoid Increasing	202	200	200	200
	Chaotic Inertia Weight	225	230	231	230

Total area scanned	Constant	95.46	98.03	98.82	98.91
	Random Inertial	95.68	98.01	98.78	98.86
	Squared Decreasing	95.15	97.84	98.70	98.80
	Sigmoid Increasing	95.51	97.58	98.53	98.64
	Chaotic Inertia Weight	95.22	97.74	98.50	98.56

*Details of the mentioned inertial functions are given in Table V

TABLE III. HYPERPARAMETERS USED FOR SIMULATION

Hyperparameter	Value
R1	Rand (0,1)
R2	Rand (0,1)
C1	0.25
C2	0.25
C3	1
C4	0.3
C5	0.2
D_{safe}	15m
N	20

Hyperparameter	Value
D_{wap}	20m
D_C	500m
D_{max}	2.5*FOV
N_s	5
α	105°
v_{max}	15m/s
v_{min}	0m/s
ω	Squared decreasing

TABLE IV. PERFORMANCE METRICS EVALUATED FOR DIFFERENT APPROACHES WITH DIFFERENT FOV'S

Metrics	Method	FOV			
		55x32	70x40	90x52	140x80
Targets Detected	Exhaustive Search	19	**20**	**20**	**20**
	Probability Graph	**20**	**20**	**20**	**20**
	Our Approach	**20**	**20**	**20**	**20**
Payload Dropped	Exhaustive Search	**18**	17	18	19
	Probability Graph	**18**	18	18	18
	Our Approach	**18**	**19**	**20**	**20**
Time for mission	Exhaustive Search	393	402	401	412
	Probability Graph	288	296	301	278
	Our Approach	**204**	**203**	**200**	**207**
Total area scanned	Exhaustive Search	99.91	99.94	99.98	99.98
	Probability Graph	98.43	98.20	98.73	99.42
	Our Approach	**94.61**	**96.84**	**97.38**	**98.32**

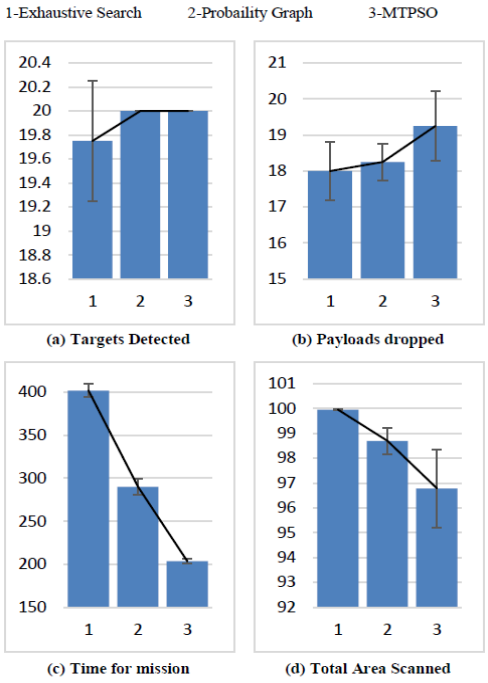

Fig. 3. Performance metrics for MTPSO, exhaustive search approach and probability graph approach

MTPSO offers a significant improvement over the other approaches in terms of response time. The complete mission time is almost half in comparison to Exhaustive Search. The algorithm is able to converge on multiple targets at a much faster rate and correctly identifies the most probable regions. It scans the least area in comparison to other approaches which is a direct consequence of discarding regions with low target probability. The computational power required for MTPSO is much less than the probability graph approach and it is easily implementable in a decentralized form on very light hardware platforms. Simulation videos can be found at link given in Reference [16].

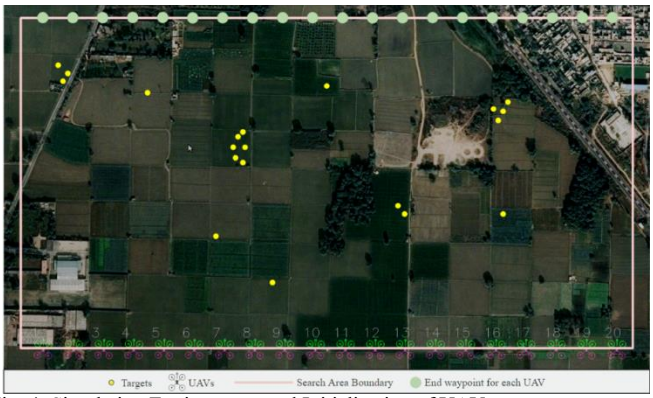

Fig. 4. Simulation Environment and Initialization of UAVs

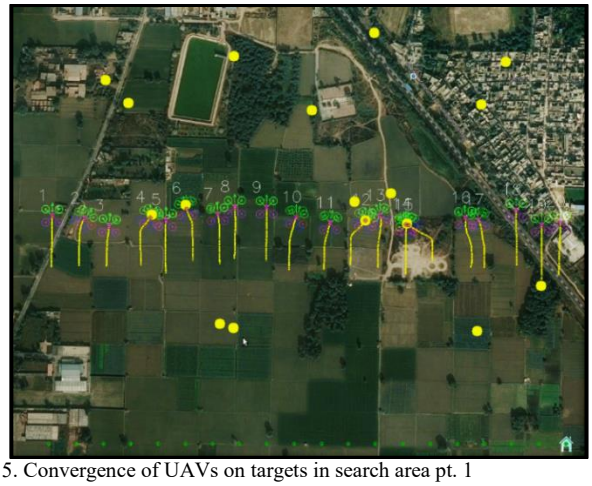

Fig. 5. Convergence of UAVs on targets in search area pt. 1

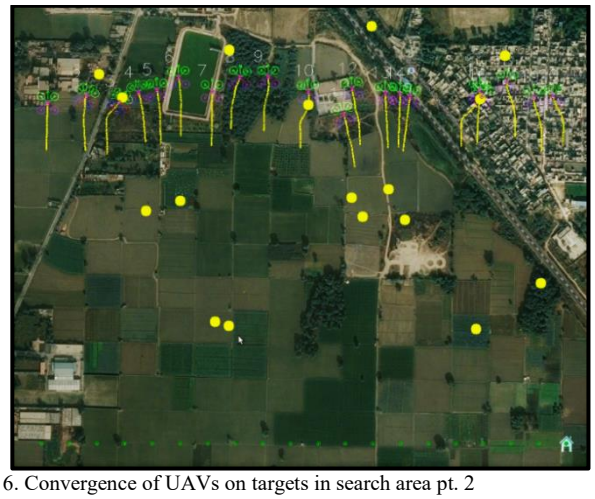

Fig. 6. Convergence of UAVs on targets in search area pt. 2

TABLE V. Different Inertial Functions Used for Testing

Function	Mathematical Form	Advantage
Constant Function	$\omega = 0.7$	1. Facilitates Global search and Manual control over the exploration-exploitation criteria
Random Inertial Weight	$\omega = 0.5 + \frac{Rand(\ )}{2}$	1. Continuously exploits the data, ideal for situations with large numbers of targets.
Squared Decreasing	$\omega = w_{max} - \left(w_{max} - w_{min} \right) \cdot \left(\frac{iteration}{iteration_{max}} \right)^2$	1. High exploration in the beginning which gradually decreases over time. 2. Ideal for moderate number of targets
Sigmoid Increasing	$\omega = \frac{\left(w_{start} - w_{end} \right)}{\left(1 + e^{10(\log(gen) - 2) \cdot (k - n \cdot gen)} \right)} + w_{end}$	1. High exploration till no targets are found, exploration decreased when targets appear. 2. Ideal for small number of targets.
Chaotic Inertia Weight	$z = 4 \cdot z(1 - z)$ $\omega = \left(w_{start} - w_{end} \right) \cdot \frac{iteration_{max} - iteration}{iteration_{max}} + w_{end} \cdot z$	1. High exploitation when targets are found. Results in faster convergence to targets. 2. Ideal in cases where immediate response is required.

V. CONCLUSION

Through this paper, we address the problem of multi-target search in an unknown environment and optimized payload delivery for rapid response scenarios. Our decentralized PSO based controller and complete system architecture* allows a single operator on the Ground Control Station to control and monitor the entire swarm. It offers two major advantages which results in better performance over traditional approaches.

First, it provides rapid response to the detected targets and can be very useful in Disaster-Relief scenarios where human lives are at stake. Second, the UAVs follow a defined structure and aim to find target clusters which increases the intensity of regional searching.

The algorithm is tested in several experiments in various simulated scenarios with varying hyperparameters where it performs with high efficiency. Our future work aims to further improve the algorithm and study the effects of varying the value of FFOV over time. Further, we also want to investigate the efficiency of the same approach in multi-target tracking.

REFERENCES

[1] Cabreira, T.M.; Brisolara, L.B.; Ferreira Jr., P.R. Survey on Coverage Path Planning with Unmanned Aerial Vehicles. Drones 2019, 3, 4.

[2] L. Bertuccelli and J. How, “Robust UAV search for environments with imprecise probability maps,” in Proc. IEEE Conf. Decision Control, Dec. 2005, pp. 5680–5685.

[3] Y. Yang, A. Minai, and M. Polycarpou, “Decentralized cooperative search by networked UAVs in an uncertain environment,” in Proc. Amer. Control Conf., Jun.–Jul. 2004

[4] J. Hu, L. Xie, K. Lum and J. Xu, "Multiagent Information Fusion and Cooperative Control in Target Search," in IEEE Transactions on Control Systems Technology, vol. 21, no. 4, pp. 1223-1235, July 2013

[5] J. Kennedy and R. Eberhart. Particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks, pages 1942-1948, IEEE Press, Piscataway, NJ, 1995

[6] Y. Shi and R. Eberhart., “A modified particle swarm optimizer”, In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, pages 69–73.

[7] IEEE, 2002. R.C. Eberhart and Y. Shi., “Tracking and optimizing dynamic systems with particle swarms”, In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, volume 1, pages 94–100.

[8] IEEE, 2002. A.Nikabadi, M.Ebadzadeh , “Particle swarm optimization algorithms with adaptive Inertia Weight : A survey of the state of the art and a Novel method”, IEEE journal of evolutionary computation , 2008

[9] R.F. Malik, T.A. Rahman, S.Z.M. Hashim, and R. Ngah, “New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight”, International Journal of Computer Science and Security (IJCSS), 1(2):35, 2007.

[10] J. Xin, G. Chen, and Y. Hai., “A Particle Swarm Optimizer with Multistage Linearly-Decreasing Inertia Weight”, In Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on, volume 1, pages 505–508. IEEE, 2009.

[11] Y. Feng, G.F. Teng, A.X. Wang, and Y.M. Yao., “Chaotic Inertia Weight in Particle Swarm Optimization”, In Innovative Computing, Information and Control, 2007. ICICIC’07.

[12] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, “Flying AdHoc Networks (FANETs): A survey,” Ad Hoc Networks, vol. 11, no. 3.

[13] Anuj Agrawal, Aniket Gupta, Joyraj Bhowmick, Anurag Singh, Raghava Nallanthighal, “A Novel Controller of Multi-Agent System Navigation and Obstacle Avoidance”, Procedia Computer Science, Volume 171, 2020, Pages 1221-1230

[14] https://www.youtube.com/watch?v=rZYvWocTQGY