

Path Planning and Network Optimization for UAV Swarms for Multi-Target Tracking

Shawon Dey, Hans D. Mittelmann, and Shankarachary Ragi, *Senior Member, IEEE*

Abstract—This paper focuses on the development of decentralized collaborative sensing and sensor resource allocation algorithms where the sensors are located on-board autonomous unmanned aerial vehicles. We develop these algorithms in the context of single-target and multi-target tracking applications, where the objective is to maximize the tracking performance as measured by the mean-squared error between the target state estimate and the ground truth while minimizing the energy costs. The tracking performance depends on the quality of the target measurements made at the sensors, which depends on the relative location of the sensors with respect to the targets. Our goal is to control the motion of the swarm of vehicles with on-board sensors to maximize the target tracking performance. Each sensor generates local noisy measurements of the target location, and the sensors maintain and update target state estimates via Bayesian data fusion rules using local measurements and the information received from the neighboring sensors. The quality of the data fusion depends on the network graph over which the sensors exchange information and the relative distance between sensors, and these determine the overall target tracking performance. For the case of multi-target tracking scenario, we also introduce sensor assignment graph in order to allocate the sensors to appropriate targets and maximize the overall tracking performance. We also assume that each sensor is powered by a limited energy source; which we assume is drained by how frequently sensors exchange information. The goal of our study is to optimize the collective motion of the vehicles/sensors (also determines the network graph connectivity and sensor assignment graph connectivity for multi-target tracking) such that the mean-squared target tracking error and the network energy costs are jointly minimized. This problem belongs to a class of hard optimization problems called *conflicting objective limited resource optimization* (COLRO). We develop fast heuristic algorithms, using dynamic programming principles, to solve this COLRO problem in real-time using a numerical optimization solver called *Knitro*, and we evaluate its performance against a widely used *particle swarm optimization* approach.

I. INTRODUCTION

There is a growing interest in decentralized and distributed autonomous sensing [1], [2], [3] and sensor allocation [4] methods, where the network connecting the sensors may be time-varying. With increasing number of sensor and surveillance systems in public places, there is a need for decentralized

S. Dey and S. Ragi are with the Department of Electrical Engineering, South Dakota School of Mines and Technology, Rapid City, SD 57701, USA: shawon.dey@mines.sdsmt.edu, shankarachary.ragi@sdsmt.edu.

H. D. Mittelmann is with the School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85287, USA: mittelmann@asu.edu.

This work was supported in part by Air Force Office of Scientific Research under grant FA9550-19-1-0070. This paper was presented in part at the Proceedings of the 2019 IEEE National Aerospace Electronics Conference (NAECON), Dayton, OH, July 15–19, 2019, pp. 88–91.

methods [5], [6], to track moving targets (e.g. movement of an intruder, movement of enemy tanks in battle field, patrolling an area) with a network of sensors. However, the decentralized collaborative sensing [7], [8] in a wireless multi-sensor network is a challenging problem, especially when there are network energy costs involved. Since the battery-powered sensor nodes have limited energy and computing resources, there is a need for methods that can trade off between the target tracking performance and the energy costs of acquiring the measurements and sharing them (with peers) over a network. Furthermore, recent development in tracking systems make it possible to deploy a large number of sensors to track multiple targets and monitor large areas. However, the proper allocation of the sensors to targets and the collaboration [9] between them in order to obtain satisfactory tracking performance with the minimization of the energy costs is more challenging.

In this study, we assume that the sensors are located on-board unmanned aerial vehicles (UAVs), where the goal is to optimize the motion controls of the vehicles for target tracking [10]. Such studies have been carried out in the past for various applications: format control [11], [12], industrial inspection [13] and remote sensing [14]. If a distributed set of autonomous vehicles are connected via a wireless network (vehicle is considered a wireless node), due to the movement of the vehicles, the links in the network graph may form and break as the relative distances between the nodes change over time, thus leading to a time-varying graph. Optimal control of UAVs over such time-varying network graphs is particularly challenging when the UAVs are performing various tasks over the network including information passing for data fusion and for cooperative optimization of motion controls. Further, if a swarm of UAVs is deployed over a large area to track multiple moving targets, the tracking performance of these UAVs may degrade with the increasing distances between the UAVs and the targets. In this type of scenario, it is highly effective to divide a swarm of UAVs into different smaller groups and control their motion by assigning them to appropriate targets. In this regard, we develop a sensor-target assignment method, which designs a graph to represent the assignment of the sensors to the targets for multi-target tracking, where the links in this graph may form and break depending on the relative motion of the UAVs with respect to that of the targets. In addition, as swarm-based systems [15] tend to have a large number of vehicles, optimizing each motion control variable in centralized manner may lead to computationally expensive optimization problems. To address this challenge, we develop multi-tier optimization strategies, where we first optimize the

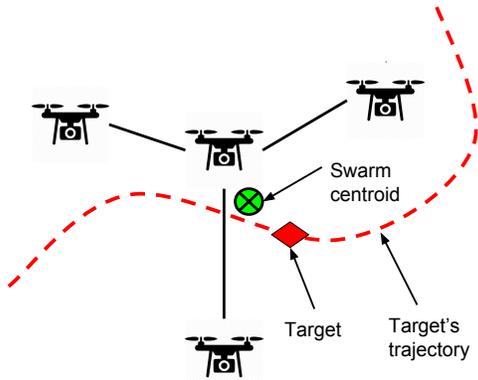


Fig. 1. Autonomous vehicle swarm tracking a target while jointly minimizing the tracking error and the energy consumption.

centroid of the UAV locations, which is then translated to the individual motion controls of the UAVs, as depicted in Figure 1. Depending on the scenario of the monitoring area, once a desired centroid and the sensor-sensor network and sensor-target assignment graphs are obtained, the vehicle controls would then be optimized to achieve the desired centroid, the sensor-sensor network and the sensor-target assignment graphs. While incorporating the above strategies, we develop a stochastic decision optimization framework to control the motion of a swarm of autonomous UAVs to track multiple moving targets, where the UAV swarm is connected via a wireless network graph, and develop approximation strategies for real-time implementation.

As mentioned earlier, we also optimize the network graph of the swarm, which determines how well the sensors (on-board the vehicles) fuse their local sensor measurements with the measurements received from the neighboring sensors, as depicted in Figure 1. Clearly, the objectives of maximizing the tracking performance and minimizing the network energy costs are competing, i.e., emphasizing one objective deteriorates the other. We refer to these problems as *competing objective limited resource optimization* (COLRO) problems. In this study, we focus on solving COLRO problems in real-time in the context of UAV swarm control for single and multi-target tracking applications. Since the network and sensor assignment graphs contain discrete variables, the above COLRO problem is a mixed integer program. To solve this optimization problem, we use a commercial solver called *Knitro* and a standard global optimization approach called *Particle Swarm Optimization* (PSO).

A. Literature Review and Key Contributions

In the context of sensor scheduling for target tracking, *partially observable Markov decision process* (POMDP) [16] and *adaptive dynamic programming* [17] approaches have been used for optimizing tracking performance and energy consumption. The authors of [18] have proposed a distributed energy optimization method for target tracking in wireless sensor networks. A study was conducted in [19] to optimize

the target tracking performance by proposing a decentralized sensor selection method. In [20], the authors proposed a formulation of the sensor management problem based on the posterior Cramer-Rao lower bound for optimizing the tracking performance and energy consumption. Furthermore, an optimal cost allocation strategy [21] is developed for both centralized and decentralized system. A study [22] has been conducted to minimize the overall network cost by incurring a cost on each node which depends on the decision vector of the respective node and also incurring additional cost on each link between two nodes, where all nodes cooperate to minimize the overall network cost. The authors of [23] optimized target detection and tracking performance, map coverage, and network connectivity for a swarm of UAVs by using Dual-Pheromone Clustering Hybrid Approach. In [24], the authors proposed a decentralized multi-target tracking system which incorporates a clustering algorithm, an optimal sensor manager, and an optimal path planner. For the case of multi-target tracking, distance-based approaches [25], [26] have been developed to optimize the sensor management system. In addition, different types of optimization techniques such as particle swarm optimization [27], [28], [29], ant colony optimization [30], reinforcement learning [31] have been used to optimize the decentralized sensing system. Despite all of these existing efforts, there is still a significant knowledge gap in terms of solving problems with competing objectives for real-time applications. This paper addresses this knowledge gap to an extent via the following *key contributions*:

- 1) We develop a fast heuristic algorithm, using dynamic programming principles, to solve the COLRO problem in near real-time.
- 2) We prove that the optimized solution of the COLRO problem is *pareto optimal*.
- 3) We develop an optimization framework to control the motion of a swarm of UAVs to track multiple targets.
- 4) We compare the performance of the *Knitro* optimization solver against particle swarm optimization to solve the above COLRO problems.

II. PROBLEM SPECIFICATION AND APPROACH

Let k represent the discrete time index. A target (e.g., ground vehicle) moves on a 2-D plane according to the *constant velocity* model [32]. Let χ_k represent the state of the target at time k , which includes its location, velocity, and acceleration. According to the *constant velocity* model, the state of the target evolves according to the following equation:

$$\chi_{k+1} = F\chi_k + v_k, \quad v_k \sim \mathcal{N}(0, Q)$$

where F is the state-transition matrix, v_k is the process noise, which is drawn from a zero-mean normal distribution with the co-variance matrix Q . Let n represent the number of UAVs in the swarm. We assume that each vehicle in the swarm has an on-board sensor that generates noisy measurements of the

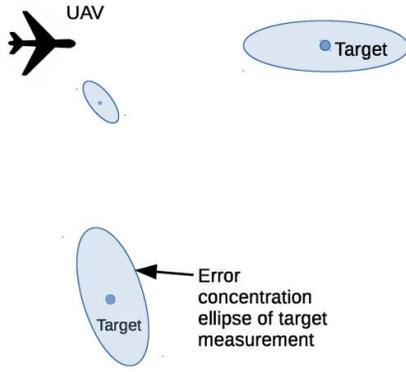


Fig. 2. Measurement error model.

target's location. The vehicles in the network are connected by a time-varying graph, represented by \mathcal{G}_k , where

$$\mathcal{G}_k = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & 0 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & 0 \end{bmatrix}$$

$a_{ij, i \neq j} = 1$ represents the ability of the sensors i and j to exchange measurements for data fusion at time k , and $a_{ij, i \neq j} = 0$ otherwise. Let \mathcal{C}_k represent the centroid of the swarm at time k . We assume that the presence of a link between two sensors at time k lets the sensors exchange local measurements (generated at time k) for data fusion purpose. The sensors on-board the vehicles generate noisy measurements of the target positions in each time step. We implement the Kalman filter to track the target state.

A. Decentralized Optimization Framework to Solve COLRO Problems

Since the swarm is a decentralized system, each vehicle runs a local target tracking algorithm (Kalman filter), which is updated using the measurements generated locally and received from the neighboring nodes, where the measurement at i th sensor is given by:

$$z_k^i = H_{\text{pos}} \chi_k + w_k^i, \quad w_k \sim \mathcal{N}(0, R_k(s_k^i, \chi_k)), \quad (1)$$

where H_{pos} is a matrix that captures just the position information in the target state vector χ_k , w_k is the measurement noise, and s_k^i is the position of the i th vehicle. We assume that the angular uncertainty is better than the range uncertainty; which is captured in the definition of the covariance matrix R_k , also captured in Figure 2. The state of the tracking algorithm is given by (ξ_k^i, P_k^i) , where ξ_k^i and P_k^i represent the mean vector and the error covariance matrix corresponding to target state estimation at the i th sensor.

Let $f_{\text{track}}(\mathcal{G}_k, \mathcal{C}_k)$ and $f_{\text{energy}}(\mathcal{G}_k, \mathcal{C}_k)$ be functions representing target tracking error (mean-squared error between

the target state) and the energy consumed respectively from sensor i 's perspective, as defined below:

$$\begin{aligned} f_{\text{track}}(\mathcal{G}_k, \mathcal{C}_k) &= \|\chi_k - \xi_k^i\|_2^2 \\ f_{\text{energy}}(\mathcal{G}_k, \mathcal{C}_k) &= \sum_i \sum_j \mathcal{G}_k(i, j) \text{linkcost}(i, j) \end{aligned} \quad (2)$$

where $\text{linkcost}(i, j)$ represents the cost of using the link between sensors i and j for data fusion purpose. For simplicity, we assume the link cost is a constant and does not depend on i and j . As this is a decentralized system, each sensor/UAVs in the system evaluates these functions using their own local target state estimates. We assume that the sensors/UAVs are aware of the positions of the other sensors/UAVs in the systems.

The goal of this study is to optimize the variables \mathcal{G}_k and \mathcal{C}_k such that the objectives f_{track} and f_{energy} are jointly minimized over a long planning horizon H . In other words, the goal boils down to solving a COLRO problem as described below:

$$\min_{\mathcal{G}_k, \mathcal{C}_k, k=0, \dots, H-1} \sum_{k=0}^{H-1} \mathbb{E}[p f_{\text{track}}(\mathcal{G}_k, \mathcal{C}_k) + (1-p) f_{\text{energy}}(\mathcal{G}_k, \mathcal{C}_k)] \quad (3)$$

where $\mathbb{E}[\cdot]$ is the expectation, and p is a weighting parameter. The above optimization problem resembles a *long-horizon optimal control* problem. These problems are notorious for high computational complexities, especially due to the presence of $\mathbb{E}[\cdot]$, which is hard to evaluate explicitly. To overcome these computational issues, a class of approximation techniques called *approximate dynamic programming* (ADP) approaches are used. With this motivation, we adopt an ADP approach called *nominal belief-state optimization* (NBO) [32], which allows us to approximate the expectation making its evaluation tractable.

1) *NBO approximation method*: Although there are several ADP approaches in the literature, e.g., policy rollout [33], we use NBO method because it is computationally less burdensome than the other methods. According to the NBO approach, the expectation is approximated by replacing the "future" noise variables with their *nominal* or mean values from the probability distributions they are drawn from. Since we model the noise variables as zero-mean Gaussian, these nominal values are zeros. After the approximation, the COLRO problem reduces to

$$\min_{\mathcal{G}_k, \mathcal{C}_k, k=0, \dots, H-1} \sum_{k=0}^{H-1} [p \tilde{f}_{\text{track}}(\mathcal{G}_k, \mathcal{C}_k) + (1-p) \tilde{f}_{\text{energy}}(\mathcal{G}_k, \mathcal{C}_k)] \quad (4)$$

where \tilde{f}_{track} and $\tilde{f}_{\text{energy}}$ are deterministic approximations to f_{track} and f_{energy} obtained from the NBO method.

The reduced COLRO problem in Eq. 4, despite being computationally more efficient to evaluate compared to the original COLRO problem, is still highly nonlinear and non-convex, and also a mixed integer program since \mathcal{G}_k contains

integer variables. We use a numerical optimization solver called *Knitro*, which allows solving mixed integer programs such as the above reduced COLRO problem with nonlinear and non-convex objective functions.

With the NBO approach, $\tilde{f}_{track}(\mathcal{G}_k, \mathcal{C}_k)$ is given by the trace of the error covariance matrix corresponding to the target state estimate, which is obtained by running the Kalman filter by assuming: 1) the future process and measurement noise variables as zero; 2) the data fusion rules are applied according to the network graph state \mathcal{G}_k .

2) *Evaluation of Optimal UAV Kinematic Controls*: The decision variables \mathcal{G}_k and \mathcal{C}_k depend on the positions of the UAVs over time. Of course, once the optimal values for \mathcal{G}_k and \mathcal{C}_k are evaluated in Eq. 4, we still need to achieve the desired graph state and the desired swarm centroid by appropriately controlling the motion of the UAVs. Since the UAV kinematic control decisions depend on the optimal values of \mathcal{G}_k and \mathcal{C}_k , we introduce a hierarchical approach with two levels, where \mathcal{G}_k and \mathcal{C}_k are optimized in the upper level (by solving Eq. 4) and the UAV kinematic controls are optimized in the lower level according to the following artificial potential field approach [34].

Let \mathcal{G}_k^* and \mathcal{C}_k^* be the optimized network graph and the centroid location at time k . Next, on each UAV we apply an attractive potential field with the center at \mathcal{C}_k^* , another attractive potential field between UAVs i and j ($j \neq i$) if $\mathcal{G}_k^*(i, j) = 1$ and the repulsive field otherwise. These two potential fields allow the UAVs to approach the desired centroid while forming/breaking network links to achieve \mathcal{G}_k^* . In addition, we also apply short-range repulsive potential fields between each pair of UAVs to avoid collisions.

3) *Proof of Pareto optimality*: Pareto optimality, in multi-objective optimization problems such as our COLRO problems, is a situation that can not be improved with respect to any objective without degrading the other objective. There may exist infinitely many Pareto-optimal solutions of the optimization problem in Eq. 4. In this section, we prove (by contradiction) that the solution to the optimization problem in Eq. 4 is Pareto-optimal. Let O be a set of all feasible solutions to the optimization problem in Eq. 4.

Let o_1 be the optimal solution to Eq. 4, and let us assume that o_1 is not Pareto-optimal. This implies there exists a feasible solution $o_2 \in O$ such that either of the two following conditions is satisfied:

- 1) $\tilde{f}_{track}(o_2) \leq \tilde{f}_{track}(o_1)$,
 $\tilde{f}_{energy}(o_2) < \tilde{f}_{energy}(o_1)$
- 2) $\tilde{f}_{track}(o_2) < \tilde{f}_{track}(o_1)$,
 $\tilde{f}_{energy}(o_2) \leq \tilde{f}_{energy}(o_1)$

If the first condition is true, then for any $p \in [0, 1)$, the following inequalities hold:

$$p\tilde{f}_{track}(o_2) \leq p\tilde{f}_{track}(o_1) \quad (5)$$

$$(1-p)\tilde{f}_{energy}(o_2) < (1-p)\tilde{f}_{energy}(o_1) \quad (6)$$

Combining Eq. 5 and Eq. 6, we get

$$p\tilde{f}_{track}(o_2) + (1-p)\tilde{f}_{energy}(o_2) < p\tilde{f}_{track}(o_1) + (1-p)\tilde{f}_{energy}(o_1) \quad (7)$$

which contradicts the assumption that o_1 is the optimal solution to Eq. 4. When $p = 1$, given the first condition is true, the following holds:

$$p\tilde{f}_{track}(o_2) + (1-p)\tilde{f}_{energy}(o_2) \leq p\tilde{f}_{track}(o_1) + (1-p)\tilde{f}_{energy}(o_1) \quad (8)$$

which is true if and only if $o_2 = o_1$. Therefore, if the first condition is true, in either cases $p \in [0, 1)$ and $p = 1$, we come to the same conclusion that o_1 is Pareto-optimal. Using similar arguments, we can also show that o_1 is Pareto-optimal when the second condition is true.

B. COLRO Optimization Framework for UAV Swarm Control for Multi-Target Tracking

In this section, we develop multi-tier optimization strategies for UAV swarm control for multitarget tracking. In the first step, we optimize the sensor-target assignment graph. Next, we optimize the centroid locations of the swarm of UAVs and the sensor-sensor network graph. Let m represent the number of targets and n the number of vehicles in the swarm, where $n \geq m$ and each vehicle has an on-board sensor to generate the position measurements of the targets. The state of the targets is given by $\chi_k = \{\chi_k^1, \chi_k^2, \dots, \chi_k^m\}$, where χ_k^l represent the state of the l th target. The state of the l th target (χ_k^l) includes its 2-D position coordinates (x_k, y_k) , its velocities (v_k^x, v_k^y) and accelerations (a_k^x, a_k^y) in x and y directions, i.e., $\chi_k^l = [x_k, y_k, v_k^x, v_k^y, a_k^x, a_k^y]$. The state of the tracking algorithm computed at i th sensor is $\xi_k^i = \{\xi_k^{i,1}, \xi_k^{i,2}, \dots, \xi_k^{i,m}\}$ and $P_k^i = \{P_k^{i,1}, P_k^{i,2}, \dots, P_k^{i,m}\}$, where $\xi_k^{i,l}$ and $P_k^{i,l}$ represent the mean vector and the error covariance matrix corresponding to the state estimation of l th target computed at i th sensor via the Kalman filter.

Since there are multiple targets, optimizing the swarm centroids and sensor-target assignment graph for each target is computationally burdensome. We develop a clustering approach to reduce this computational complexity, where the targets are clustered into smaller groups depending on their relative distances. Each sensor/UAV runs this clustering algorithm, where the two targets are clustered if the perceived distance between these targets (obtained using the local state estimates of the targets) is less than or equal to a specific threshold. Furthermore, the swarm of UAVs is also divided into multiple smaller groups, where each group follows a specific cluster of targets. Also, the optimal division of the swarm into smaller groups is integrated within optimization problem to optimize the sensor-target assignment graph.

To determine the target clusters, we need to evaluate the distance between all targets. Since the target location is known only with uncertainty, we can not evaluate the exact Euclidian distance between two targets. Therefore, we use Bhattacharyya distance [35] to measure the statistical distance between the probability distributions of the targets.

Let B_k^i matrix capture the statistical distance between all targets at time k from sensor i 's perspective,

$$B_k^i = \begin{bmatrix} 0 & b_{12} & \dots & b_{1m} \\ b_{21} & 0 & \dots & b_{2m} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & 0 \end{bmatrix}$$

where b_{lh} represents the Bhattacharyya distance between targets l and h , as defined in [35] and repeated here for completeness:

$$b_{lh} = \frac{1}{8} (\xi_k^l - \xi_k^h)^T \left(\frac{P_k^l + P_k^h}{2} \right)^{-1} (\xi_k^l - \xi_k^h) + \frac{1}{2} \log \left(\frac{\det \left(\frac{P_k^l + P_k^h}{2} \right)}{\sqrt{\det(P_k^l) \det(P_k^h)}} \right) \quad (9)$$

where, ξ_k^l and P_k^l represent the mean vector and the error covariance matrix corresponding to the state estimation of l th target, \det represents the matrix determinant operator.

Let C_k^i matrix represent the assignment of targets to clusters from sensor i 's perspective, where

$$C_k^i = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1N_i} \\ t_{21} & t_{22} & \dots & t_{2N_i} \\ \dots & \dots & \dots & \dots \\ t_{m1} & t_{m2} & \dots & t_{mN_i} \end{bmatrix}$$

$t_{lc} = 1$ means that the target l is belonged to the cluster of targets c at time k , and $t_{lc} = 0$ otherwise. Also, N_i is the total number of target clusters from sensor i 's perspective and m is the total number of targets which is denoted earlier. Each target is clustered with the closest neighboring target if the statistical distance between the two targets is less than or equal to a certain threshold R . A target is added to an existing cluster if it is within the R distance of any target of the cluster. Each sensor in the system evaluate this clustering approach and update the matrix C_k^i .

Furthermore, as we mentioned earlier, the swarm of UAVs deployed in the system is divided into multiple smaller groups based on the sensor-target assignment graph. Let the sensor/UAV i be assigned to the clusters of targets by an assignment index \mathcal{A}_k^i , where

$$\mathcal{A}_k^i = [b_{i1} \quad b_{i2} \quad \dots \quad b_{iN_i}]$$

$b_{ic} = 1$ means the sensor i is assigned to cluster c at time k , and $b_{ic} = 0$ otherwise. The graph \mathcal{A}_k^i is subject to a constraint that the sensor i gets assigned to at least one cluster of targets at time k . Each sensor optimizes this graph based on the relative distance from the sensor to the clusters of targets and the total number of targets in each cluster. Let D_k^i is a set of distances which represent the statistical distance between sensor i and all targets as described below:

$$D_k^i = [w_{i1} \quad w_{i2} \quad \dots \quad w_{im}]$$

where, w_{il} represent the distance between sensor i and target l .

Let W_k^{ic} be a weight parameter associated with cluster c from sensor i 's perspective at time k defined as

$$W_k^{ic} = \frac{m}{s_k^{ic}}$$

s_k^{ic} represents the total number of targets in cluster c at time k , and m represents the total number of targets deployed in the monitoring area. This weight parameter defines the size of each cluster. Each sensor considers the average distance of the targets and the size of each cluster while choosing a cluster to follow and evaluate the following function:

$$f(\mathcal{A}_k^i) = \sum_{c=1}^{N_i} \sum_{l=1}^m \mathcal{A}_k^i(c) C_k^i(l, c) D_k^i(l) W_k^{ic} (s_k^{ic})^{-1} \quad (10)$$

Sensor i optimizes the variable \mathcal{A}_k^i by minimizing the function $f(\mathcal{A}_k^i)$ over a long planning horizon H as described below:

$$\min_{\mathcal{A}_k^i, k=0, \dots, H-1} \sum_{k=0}^{H-1} E[f(\mathcal{A}_k^i)] \quad (11)$$

where, $E[\cdot]$ is the expectation. After approximating the expectation $E[\cdot]$ by using the NBO approach, the above problem reduces to

$$\min_{\mathcal{A}_k^i, k=0, \dots, H-1} \sum_{k=0}^{H-1} \tilde{f}(\mathcal{A}_k^i) \quad (12)$$

where \tilde{f} are deterministic approximations to f obtained from the NBO method.

When the desired sensor-target assignment graph is achieved, each sensor optimizes the sensor-sensor network graph and the cluster's centroid (the assigned cluster for this sensor/UAV).

Furthermore, the vehicles in the network are connected by a time-varying graph, represented by \mathcal{G}_k , where

$$\mathcal{G}_k = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & 0 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & 0 \end{bmatrix}$$

$a_{ij, i \neq j} = 1$ represents the ability of the sensors i and j to exchange measurements for data fusion at time k , and $a_{ij, i \neq j} = 0$ otherwise.

As we mentioned earlier, UAV i is assigned to a specific cluster of targets based on the optimized graph \mathcal{A}_k^{i*} . Also, each UAV optimizes the centroid location of its assigned cluster of targets. Let, $Cent_k^i$ represents the centroid of the cluster of targets that sensor i is following at time k .

Let $f_{track}(\mathcal{G}_k, Cen_k^i)$ and $f_{energy}(\mathcal{G}_k, Cen_k^i)$ be functions representing target tracking error and the energy consumed respectively from sensor i 's perspective as defined below:

$$f_{track}(\mathcal{G}_k, Cen_k^i) = \sum_{c=1}^{N_i} \sum_{l=1}^m \mathcal{A}_k^{i*}(c) C_k^i(l, c) \quad (13)$$

$$\left\| \chi_k^l - \xi_k^{i,l} \right\|_2^2 (s_k^{ic})^{-1}$$

where, χ_k^l is the state vector of the l th target and $\xi_k^{i,l}$ represent the mean vector corresponding to the state estimation of l th target computed at i th sensor.

The energy cost function is given by

$$f_{energy}(\mathcal{G}_k, Cen_k^i) = \sum_q \sum_r \mathcal{G}_k(q, r) \text{linkcost}(q, r) \quad (14)$$

where, $\text{linkcost}(q, r)$ represents the cost of using the link between sensors q and r for data fusion purpose. We assume that the link cost is directly proportional to the Euclidean distance between sensor q and r with a proportionality constant ϕ .

We optimize the variables \mathcal{G}_k and Cen_k^i such that the objectives f_{track} and f_{energy} are jointly minimized over a long time horizon H . As previously described, after approximating the expectation $E[\cdot]$ by using the NBO approach, we solve the following COLRO problem as described below:

$$\min_{\mathcal{G}_k, Cen_k^i, k=0, \dots, H-1} \sum_{k=0}^{H-1} [p \tilde{f}_{track}(\mathcal{G}_k, Cen_k^i) + (1-p) \tilde{f}_{energy}(\mathcal{G}_k, Cen_k^i)] \quad (15)$$

where \tilde{f}_{track} and \tilde{f}_{energy} are deterministic approximations to f_{track} and f_{energy} obtained from the NBO method. Also, p is a weighting parameter. We use the optimization solver *Knitro* to solve the above optimization problem. The kinematic control decisions of the i th UAV depend on the optimized graph \mathcal{A}_k^{i*} and the centroid Cen_k^i . We introduced a hierarchical model described earlier in section II-A2, where \mathcal{A}_k^i , \mathcal{G}_k and Cen_k^i are optimized in the higher level, and UAV kinematic controls are optimized in the lower level according to the potential field approach. At time k , on each UAV we apply an attractive potential field with the centroid Cen_k^i . In addition, we also apply short-range repulsive potential fields between each pair of UAVs to avoid collisions.

C. Performance comparison of KNITRO against Particle Swarm Optimization

In this section, we evaluate the performance of the optimization solver *Knitro* against an established algorithm *particle swarm optimization* (PSO). Artelys *Knitro* is an optimization solver for finding solutions for both continuous and discrete optimization problem. It is efficient at solving mixed integer programs as well. We solved the optimization problem in Eq. 4 using *Knitro* as already described in Chapter 3. We also solved the reduced COLRO problem in Eq. 4 using PSO. PSO was originally introduced by Kennedy Eberhart in [36]. This

algorithm has since been used in many areas because of its simplicity and flexibility. Furthermore, various modifications of the PSO have been proposed by many researchers. In the original form, PSO is a method for finding the global minimum of an objective function, where a group of n particles searches through a domain space with dimension d . This group of particles is referred to as swarm. Since \mathcal{G}_k contains discrete variables and C_k contains continuous variables in the COLRO problems discussed in the previous chapters, we use both binary and continuous version of PSO. The steps of applying PSO in our problem are summarized as follows:

- 1) Initialize a population of particles with positions and velocities. Let n is the total number of particles. Let $x_k^i = [x_k^{i1}, x_k^{i2}, \dots, x_k^{id}]$ and $v_k^i = [v_k^{i1}, v_k^{i2}, \dots, v_k^{id}]$ are the position and velocity of the i th particle at iteration k , respectively. Both the position and velocity vectors live in \mathbb{R}^d .
- 2) Evaluate the objective function (Eq. 4) using the position of each particle. The best position associated with the best value of the objective function for the particle i is called the particle's best position and it is defined as $p_k^i = [p_k^{i1}, p_k^{i2}, \dots, p_k^{id}]$. If current value of the objective function is better than the previous value, then set particle best value equal to the current value and particle best position equal to the current position. Also global best g_k represent the best position among all particles so far and is defined as $g_k = [g_k^1, g_k^2, \dots, g_k^d]$.
- 3) Update the iteration from k to $k+1$. For continuous variables, update the velocity and position of each particle through dimension d according to the following equation:

$$v_{k+1}^{ie} = w v_k^{ie} + c_1 r_1 (p_k^{ie} - x_k^{ie}) + c_2 r_2 (g_k^e - x_k^{ie}) \quad (16)$$

$$x_{k+1}^{ie} = x_k^{ie} + v_{k+1}^{ie} \quad (17)$$

for $e = 1, 2, \dots, d$, where w is an inertia weight, which balance the global and local search ability. A large inertia weight facilitates a global search, also known as exploration, and a small inertia weight facilitates a local search, known as exploitation. However, the simulation is time consuming when the value of w is lower. c_1 (cognition learning factor) and c_2 (social learning factor) are arbitrary constants, and r_1 and r_2 are two random variables uniformly distributed in $[0, 1]$.

For discrete variables, we update the velocity using Eq. 5.1 and then introduce a sigmoid function which is used to constrain the variables within 0 or 1,

$$\text{Sig}(v_{k+1}^{ie}) = \frac{1}{1 + e^{(-v_{k+1}^{ie})}} \quad (18)$$

$$x_{k+1}^{ie} = \begin{cases} 1, & \text{if } R < \text{Sig}(v_{k+1}^{ie}) \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

where R is a random number selected from a uniform distribution in $[0,1]$.

- 4) Search is terminated if the number of iterations reaches a threshold.

III. RESULTS

A. Decentralized Optimization Framework to Solve COLRO Problems

We implement the above-discussed methods (Section II-A) of solving the COLRO problem in MATLAB for a scenario with three UAVs tracking a single target. We set the time horizon $H = 6$ and apply the *receding horizon control* [32] approach for planning and implementing the optimized decisions. For bench-marking, we also implement the centralized UAV motion planning approach discussed in [32]; we call this *centralized fusion approach*.

Figure 3 shows the trajectories of three UAVs tracking a target. The target and the UAVs begin their motion in the bottom-left region, and move toward the top-right region. Figures 4 and 5 show the network link status (three links for three UAVs) as a function of time for the weighting parameter in Eq. 4 set to $p = 0.2$ and $p = 0.01$ respectively. Clearly, in Figure 5, the UAVs exchange information less often compared to the scenario in Figure 4. These figures clearly demonstrate our ability to trade-off between the two competing performance indices smoothly. We evaluate the normed error between the actual target location (ground truth) and the target location estimate at each sensor over time for the scenario in Figure 3. In Figure 6, we compare the performance of the above-discussed approach against the *centralized fusion approach*, which clearly shows that the tracking performance of the centralized approach is just marginally better than our COLRO-based methods discussed here. Of course, in the centralized approach, the performance with respect to the network energy costs is ignored. In other words, our approach, while slightly trading off the tracking performance, gains significantly in the performance with respect to the network energy consumption.

B. COLRO Optimization Framework for UAV Swarm Control for Multi-Target Tracking

In this part of our study, we implement the method of solving COLRO problem for multi-target tracking scenarios discussed in Section II-B. Here, we set $H = 6$ and also apply *receding horizon control* [32] for planning and implementing the optimized decisions. We also implement the *centralized fusion approach* discussed earlier for benchmarking.

Figures 7 and 8 show the trajectories of UAVs tracking multiple targets in different scenarios. Figure 7 shows a scenario with nine UAVs tracking five targets. Similarly, Figure 8 shows a scenario with five UAVs tracking three targets. In each case, the targets and UAVs begin their motion from the bottom region and move towards the top in different directions. Each UAV tracks the assigned cluster of targets and also maintains network links with other UAVs deployed in the system, where the network links between UAVs depend on the

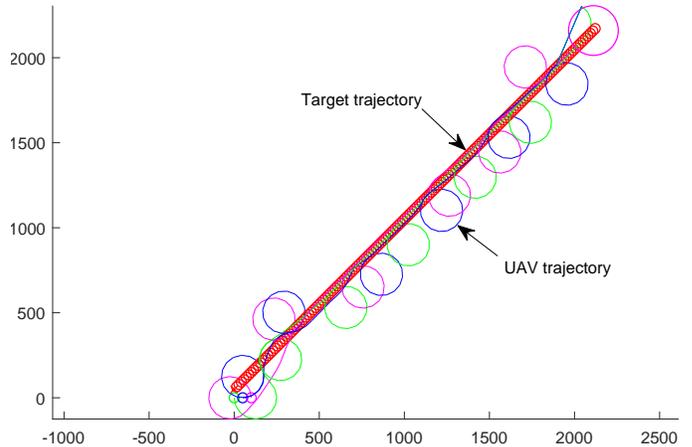


Fig. 3. Three UAVs tracking a target.

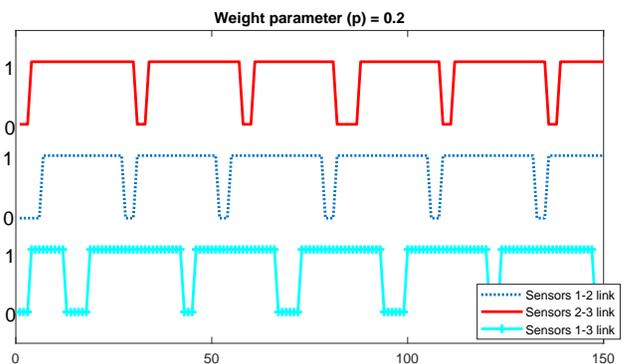


Fig. 4. Status of UAV network links over time with $p = 0.2$ (1 means active and 0 otherwise)

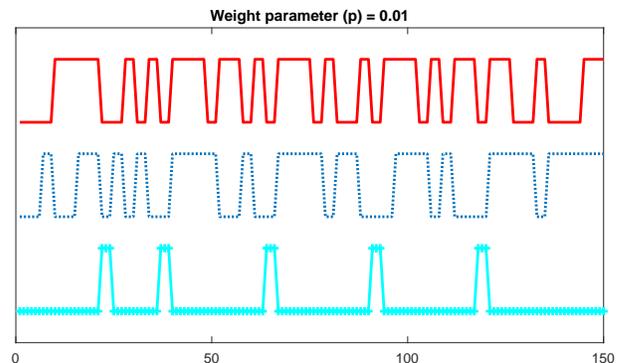


Fig. 5. Status of UAV network links over time with $p = 0.01$ (1 means active and 0 otherwise)

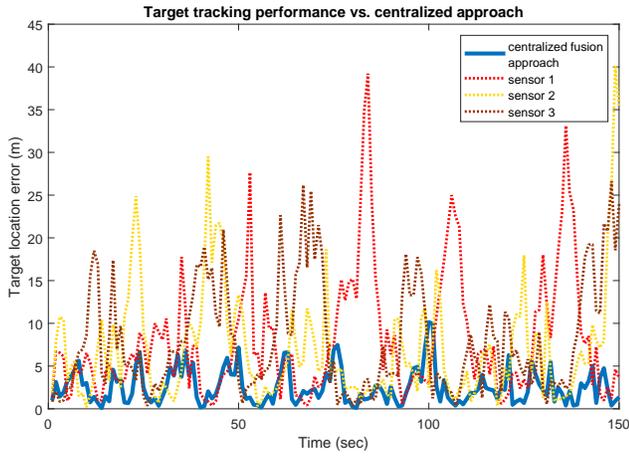


Fig. 6. Normed target location error: COLRO-based approach vs. centralized approach.

graph \mathcal{G}_k . We evaluate the normed target location estimation error at each sensor over time for the scenario in Figure 8, where all targets start at the bottom, and as the simulation progresses, two targets move toward the north-east and one target moves toward the north-west direction. The UAVs start at the bottom and move according to the control optimization methods discussed in section II-B, with controls obtained by solving the optimization problem in Eq. 12 and Eq. 15.

In figures 9, 10 and 11, we compare the performance of our approach against the centralized fusion approach. Figure 9 depicts the normed target location error for target 1, where the tracking performance of the sensors (1 and 2) assigned to target 1 is better than the performance of the other sensors. Similarly, from figures 10 and 11, it is clearly shown that the performance of tracking the targets 2 and 3 by sensors 1, 4, and 5 is better than sensors 2 and 3. The communication cost of our decentralized approach is compared with the cost of the centralized approach in Figure 12. From the above figures, it is clear that the COLRO based approach significantly reduces the communication cost by trading off the tracking performance slightly.

C. Performance comparison of KNITRO against Particle Swarm Optimization

In this section, we implement the method discussed in section II-C using *Knitro* and PSO in MATLAB.

We performed a Monte-Carlo study to compare the statistical performance of *Knitro* with PSO. We ran a scenario with three UAVs and one target with 100 Monte-Carlo runs, and in each run, we measured the average target tracking error. Figure 13 shows that *Knitro* has a significant statistical edge over PSO in terms of the target tracking performance. With this, we conclude that *Knitro* is able to solve the COLRO problems more efficiently compared to PSO in terms of target tracking performance.

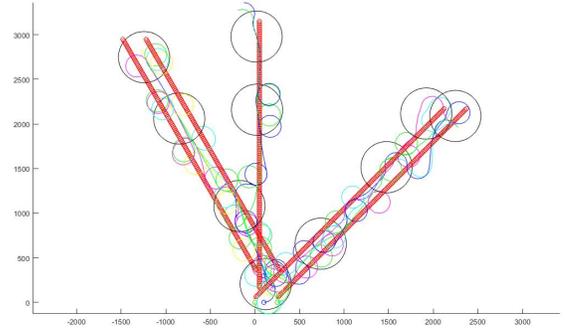


Fig. 7. Nine UAVs tracking five targets.

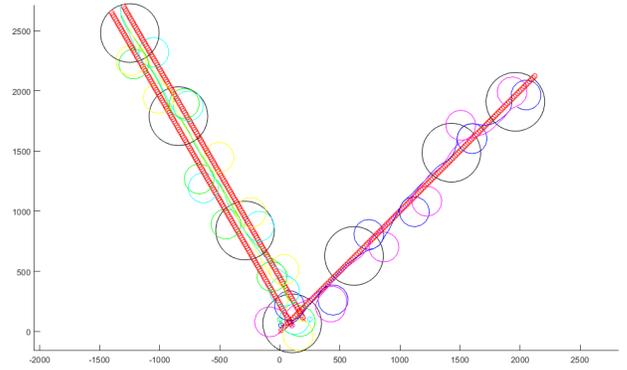


Fig. 8. Five UAVs tracking three targets.

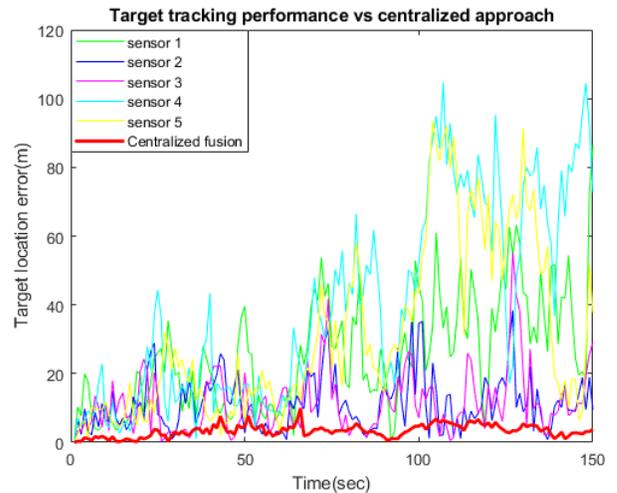


Fig. 9. Normed target location error for target 1 in Fig. 8

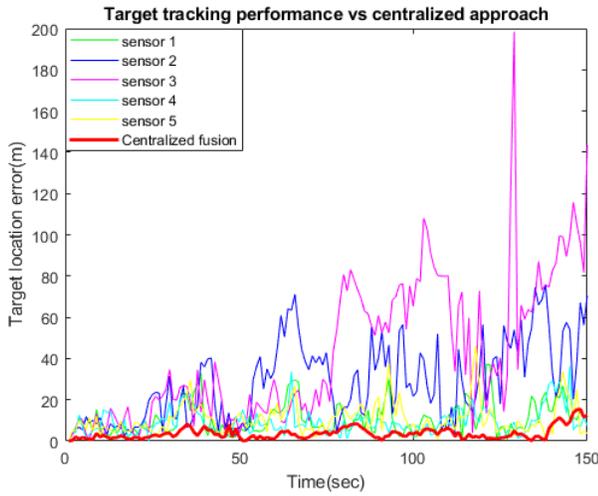


Fig. 10. Normed target location error for target 2 in Fig. 8

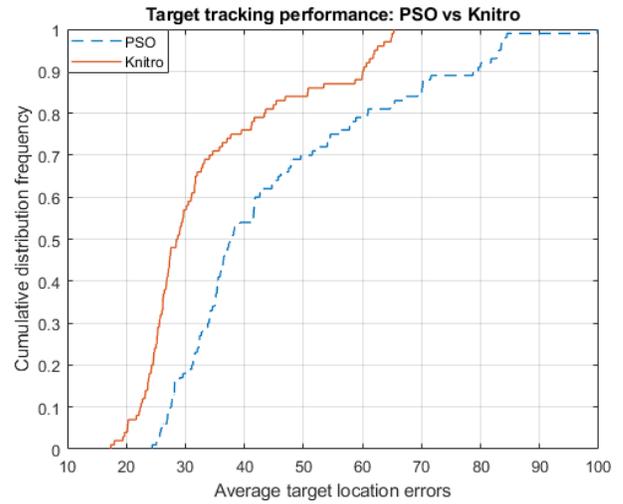


Fig. 13. Normed target location error: PSO vs Knitro

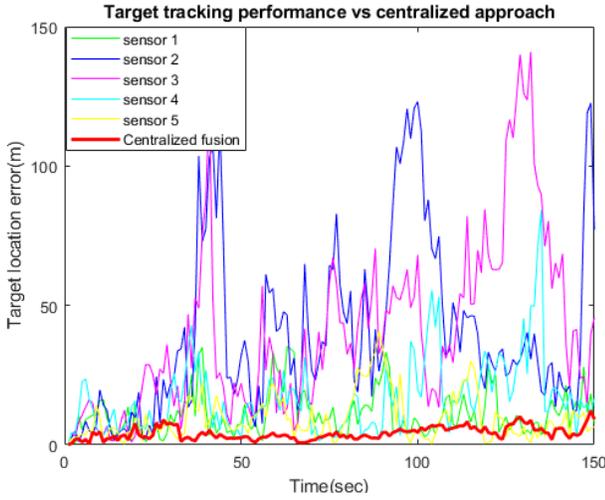


Fig. 11. Normed target location error for target 3 in Fig. 8

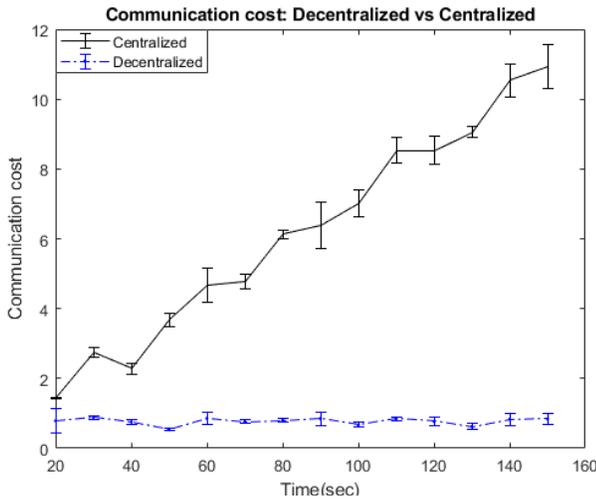


Fig. 12. Communication cost: decentralized vs centralized for Fig. 8

IV. CONCLUSIONS AND FUTURE SCOPE

In this work, we presented real-time heuristic approaches to solve a *competing objective limited resource optimization* (COLRO) problem in the context of a networked unmanned aerial vehicle (UAV)/sensors system for target tracking applications. Specifically, in our study, the objective is to optimize the motion of a swarm of UAVs (equipped with sensors) to track moving targets, while jointly minimizing the tracking error and the network energy costs. This optimization problem led to *long horizon optimal control* problem, which is known to be computationally hard. We used an approximate dynamic programming approach called *nominal belief state optimization* to solve the above COLRO problem. We solved the above problem in the context of single and multi-target tracking applications. In this study, the motion of the swarm of UAVs was controlled by optimizing the centroid location, and the network and sensor assignment graphs. We also developed clustering approaches for multi-target tracking scenarios, where targets are clustered (based on their mutual statistical distances between their probability distributions) to reduce the computational effort in the UAV-target assignment.

Furthermore, we developed UAV clustering methods, where UAVs are clustered into smaller groups depending on the number and geographical locations of the target clusters, and then the collective motion of these clusters are optimized via COLRO formulation. We tested the performance of these approaches in simulated environment (implemented in MATLAB), and compared the performance of our approaches with a *centralized fusion approaches* for benchmarking. We found our methods to lose on the tracking performance only marginally compared to the centralized fusion approach, while significantly saving the network energy costs. We used a numerical optimization solver called *knitro* to solve the above control problems formulated as COLRO. We also evaluated the performance of *knitro* against *particle swarm optimization* ap-

proach, and found the Knitro approach to perform marginally better than the PSO approach.

Although our COLRO formulation and the decentralized decision optimization methods for UAVs proved to be effective in target tracking applications, there still exists a knowledge gap in terms of how well our heuristic methods (approximate dynamic programming) perform with respect to the optimal solution. As a potential future study, one can explore the possibility of bounding the performance of the ADP schemes, under potentially weak conditions, to determine how close our approximation methods are to the optimal solution.

REFERENCES

- [1] W. Xiao, C. K. Tham, and S. K. Das, "Collaborative sensing to improve information quality for target tracking in wireless sensor networks," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010, pp. 99–104.
- [2] M. Zhang and D. Kingston, "Time-space network based exact models for periodical monitoring routing problem," in *2015 American Control Conference (ACC)*, 2015, pp. 5264–5269.
- [3] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous uav path planning and estimation," *IEEE Robotics Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.
- [4] Y. Fu, Q. Ling, and Z. Tian, "Distributed sensor allocation for multi-target tracking in wireless sensor networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3538–3553, 2012.
- [5] T. H. Chung, J. W. Burdick, and R. M. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2416–2422.
- [6] S. Ragi and E. K. P. Chong, "Decentralized guidance control of uavs with explicit op-timization of communication," *Journal of Intelligent and Robotic System*, vol. 73, no. 4, p. 811–822, 2014.
- [7] A. Ryan, J. Tisdale, M. Godwin, D. Coatta, D. Nguyen, S. Spry, R. Sengupta, and J. K. Hedrick, "Decentralized control of unmanned aerial vehicle collaborative sensing missions," in *2007 American Control Conference*, 2007, pp. 4672–4677.
- [8] T. Vercauteren and Xiaodong Wang, "Decentralized sigma-point information filters for target tracking in collaborative sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2997–3009, 2005.
- [9] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.
- [10] S. A. P. Quintero, D. A. Copp, and J. P. Hespanha, "Robust uav coordination for target tracking using output-feedback model predictive control with moving horizon estimation," in *2015 American Control Conference (ACC)*, 2015, pp. 3758–3764.
- [11] D. van der Walle, B. Fidan, A. Sutton, Changbin Yu, and B. D. O. Anderson, "Non-hierarchical uav formation control for surveillance tasks," in *2008 American Control Conference*, 2008, pp. 777–782.
- [12] I. Shames, B. Fidan, and B. D. O. Anderson, "Close target reconnaissance using autonomous uav formations," in *2008 47th IEEE Conference on Decision and Control*, 2008, pp. 1729–1734.
- [13] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, "A uav system for inspection of industrial facilities," in *2013 IEEE Aerospace Conference*, 2013, pp. 1–8.
- [14] "Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs)," *Photogrammetric Engineering Remote Sensing*, vol. 81, no. 4, 2015.
- [15] G. Varela, P. Caamaño, F. Orjales, Deibe, F. López-Peña, and R. J. Duro, "Swarm intelligence based approach for real time uav team coordination in search operations," in *2011 Third World Congress on Nature and Biologically Inspired Computing*, 2011, pp. 365–370.
- [16] G. K. Atia, V. V. Veeravalli, and J. A. Fuemmeler, "Sensor scheduling for energy-efficient target tracking in sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4923–4937, 2011.
- [17] F. Liu, W. Xiao, S. Chen, and C. Jiang, "Adaptive dynamic programming-based multi-sensor scheduling for collaborative target tracking in energy harvesting wireless sensor networks," *Sensors*, vol. 18, no. 12, p. 4090, Nov 2018. [Online]. Available: <http://dx.doi.org/10.3390/s18124090>
- [18] X. Wang, J. Ma, S. Wang, and D. Bi, "Distributed energy optimization for target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 73–86, 2010.
- [19] R. Tharmarasa, T. Kirubarajan, A. Sinha, and T. Lang, "Decentralized sensor selection for large-scale multisensor-multitarget tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1307–1324, 2011.
- [20] R. Tharmarasa, T. Kirubarajan, J. Peng, and T. Lang, "Optimization-based dynamic sensor management for distributed multitarget tracking," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 5, pp. 534–546, 2009.
- [21] E. Laz and S. Gezici, "Optimal cost allocation in centralized and decentralized detection problems," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2016, pp. 1–5.
- [22] X. Cao and K. J. R. Liu, "Distributed efficient optimization for general network cost minimization problems," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.
- [23] M. R. Brust, M. Zurad, L. Hentges, L. Gomes, G. Danoy, and P. Bouvry, "Target tracking optimization of uav swarms based on dual-pheromone clustering," in *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*, 2017, pp. 1–8.
- [24] N. Farmani, L. Sun, and D. J. Pack, "A scalable multitarget tracking system for cooperative unmanned aerial vehicles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 4, pp. 1947–1961, 2017.
- [25] L. Chen, A. I. El-Fallah, R. K. Mehra, R. P. S. Mahler, J. R. Hoffman, C. A. Stelzig, and M. G. Alford, "Multisensor management optimization for multitarget tracking based on Hausdorff distance minimization," in *Signal Processing, Sensor Fusion, and Target Recognition XI*, I. Kadar, Ed., vol. 4729, International Society for Optics and Photonics. SPIE, 2002, pp. 259 – 269. [Online]. Available: <https://doi.org/10.1117/12.477611>
- [26] J. R. Hoffman and R. P. S. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 34, no. 3, pp. 327–336, 2004.
- [27] K. K. Veeramachaneni and L. A. Osadciw, "Dynamic sensor management using multi-objective particle swarm optimizer," in *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2004*, B. V. Dasarathy, Ed., vol. 5434, International Society for Optics and Photonics. SPIE, 2004, pp. 205 – 216. [Online]. Available: <https://doi.org/10.1117/12.542927>
- [28] K. Veeramachaneni and L. Osadciw, "Swarm intelligence based optimization and control of decentralized serial sensor networks," in *2008 IEEE Swarm Intelligence Symposium*, 2008, pp. 1–8.
- [29] X. Wang, S. Wang, and J. J. Ma, "An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment," *Sensors*, vol. 7, no. 3, p. 354–370, Mar 2007. [Online]. Available: <http://dx.doi.org/10.3390/s7030354>
- [30] C. Cheng, C. K. Tse, and F. C. M. Lau, "A clustering algorithm for wireless sensor networks based on social insect colonies," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 711–721, 2011.
- [31] M. Mihaylov, K. Tuyls, and A. Nowé, "Decentralized learning in wireless sensor networks," in *Adaptive and Learning Agents*, M. E. Taylor and K. Tuyls, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 60–73.
- [32] S. Ragi and E. K. P. Chong, "Uav path planning in a dynamic environment via partially observable markov decision process," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2397–2412, 2013.
- [33] E. K. P. Chong, C. M. Kreucher, and A. O. Hero, "Partially observable markov decision process approximations for adaptive sensing," *Discrete Event Dynamic Systems*, vol. 19, no. 3, pp. 377–422, Sep 2009. [Online]. Available: <https://doi.org/10.1007/s10626-009-0071-x>
- [34] Min Gyu Park, Jae Hyun Jeon, and Min Cheol Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570)*, vol. 3, 2001, pp. 1530–1535 vol.3.
- [35] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, no. 1, pp. 52–60, 1967.

- [36] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.