Bridging Cooperative Sensing and Route Planning of Autonomous Vehicles

P. B. Sujit, Daniel E. Lucani, and João B. Sousa

***Abstract*—Autonomous Vehicles (AV) are used to solve the problem of data gathering in large scale sensor deployments with disconnected clusters of sensors networks. Our take is that an efficient strategy for data collection with AVs should leverage i) cooperation amongst sensors in communication range of each other forming a sensor cluster, ii) advanced coding and data storage techniques for easing the cooperation process, and iii) AV route-planning that is both content- and cooperation-aware. Our work formulates the problem of efficient data gathering as a cooperative route-optimization problem with communication constraints. We also analyze (network) coded data transmission and storage for simplifying cooperation amongst sensors as well as data collection by the AV. Given the complexity of the problem, we focus on heuristic techniques, such as particle swarm optimization, to calculate the AV's route and the times for communication with each sensor and/or cluster of sensors. We analyze two extreme cases, i.e., networks with and without intra-cluster cooperation, and provide numerical results to illustrate that the performance gap between them increases with the number of nodes. We show that cooperation in a 100 sensor deployment can increase the amount of data collected by up to a factor of 3 with respect to path planning without cooperation.**

***Index Terms*—Robot sensing systems, autonomous vehicles, path planning, network coding, optimization**

I. INTRODUCTION

ADVANCES in sensor, computer and communications technologies are enabling large scale sensor deployments over wide geographic areas. Sensor nodes with sensing, communication, and computation capabilities are now available in small form factors with minimal power requirements. Applications include environmental studies and military operations. The problem of data gathering in large scale deployments of sensor networks is a challenging one. The nature of the sensor data imposes communication requirements, including minimum throughput and maximum delay guarantees, that need to be accounted for in the design of data collection strategies and protocols. In addition, the communications network may be composed of disconnected clusters of sensors.

Autonomous Vehicle (AV) control and advanced communications techniques constitute key enablers towards achieving large-scale data sensing and gathering in networks with disconnected clusters of sensors. In this sense, AVs act as *ferries*, used to gather, store, and send data to and amongst sensors. Although previous work has considered AVs for data collection, the problem has been approached primarily from a vehicle control perspective where communication requirements can be met, or not, as a byproduct of that control.

Our approach is different from currently proposed methodologies to data gathering using AVs in two fundamental ways: (1) we rely on a holistic view of the problem, where communication and data gathering are the driving forces behind the control of the AVs, rather than a byproduct of the control; (2) we leverage cooperation of interconnected sensors organized in clusters and advanced coding techniques to facilitate reliable storage and availability of the data in preparation for a visit from the AV and to provide higher reliability in the data gathering process once the AV is in range.

We shall use network coding [1] as a unifying mechanism for the latter, because its encoding and decoding primitives remain the same for both transmission, e.g., [1],[2] and storage of data, e.g., [3] [4]. Network coding encourages the system to mix different data packets through coding, rather than storing and forwarding copies of packets that are routed through the network. Under this paradigm, it is no longer required for the system to keep track of which packets have been received: receivers need only aim at accumulating enough coded packets in order to recover the data. Network coding thus enables sensors to store some coded data from its neighbors, which translates in the AVs communicating with a fraction of the sensors in the cluster in order to gather the data. An AV that need not physically visit every node will reduce the time in each cluster, improving the overall system performance.

Seeking to characterize the gains expected from optimal path routing with and without sensor cooperation, we make the following contributions:

- *Fundamental Analysis:* We propose a model and optimization formulations that enable us to solve the aforementioned communications-oriented route selection and planning problem as well as studying the advantages of intra-cluster cooperation. We analyze special cases where the problem of data dissemination and route planning can be solved with minimal coupling.
- *Algorithm Design:* We propose algorithms based on particle swarm optimization to solve this problem for sensor networks with and without cooperation. Our algorithms scale very well with the number of sensor nodes, taking at most 4 seconds to calculate the path for 100 sensors.
- *Performance Evaluation and Comparison:* We compare

Manuscript received on 1 August 2011, and revised on 1 May 2012. This work was supported in part by the Fundação para Ciência e Tecnologia under projects PEst-OE/EEI/LA0008/2011 and PTDC/EEA-CRO/104901/2008. The first author is supported by the Fundação para Ciência e a Tecnologia.

P. B. Sujit and J. B. Sousa are with the Department of Electrical and Computer Engineering, Faculdade de Engenharia da Universidade do Porto, Portugal (e-mail: {sujit, jtasso}@fe.up.pt).

D. E. Lucani is with the Instituto de Telecomunicações, DEEC, Faculdade de Engenharia da Universidade do Porto, Portugal (e-mail: dlucani@fe.up.pt).

Digital Object Identifier 10.1109/JSAC.2012.120607.

the performance of our mechanisms with a sequential, a greedy, and a traveling salesman inspired data collection techniques. We also illustrate numerically the performance gap amongst networks with and without cooperation and show that our cooperation algorithms can recover up to 3 times more data than the TSP-inspired algorithm for the same mission time.

II. Related Work

Optimal route or trajectory planning has been an essential optimization problem for several decades, targeting different applications and objective functions to optimize [5]. However, route planning for unmanned vehicles, especially with kinematic constraints, has been addressed only recently [6][7]. More recently, communication-aware routing and trajectory planning has become an area of keen interest to different robotics and communication communities [8] [9] [10].

The problem of communication-aware route planning is typically used for target tracking applications with multiple robots where the objective is to design the routes for each robot in order to preserve communication connectivity between the robots [11]. Due to inherent issues with communication and NP-Hard nature of the problem, optimal route planning is still an open issue for real-time applications. Most of the solutions are developed with heuristics that optimize certain objective, e.g., communication bandwidth [8], interference [12].

Our approach aims to propose strategies for the AV to plan its route to meet certain communication and networking objectives. Hence, our problem is not purely vehicle route planning [13] and does not use communication-awareness for multiple AVs to ensure proper coordination of joint actions. Perhaps the closest work is that of Ref. [9], which studies a special case where a robot visits sensor nodes to gather data. In particular, Ref. [9] focuses mostly on strategies to shut down and wake up the sensors to attain a good trade-off between the sampling time, related to the time the robot waits for the sensor to wake-up, and the overall energy expenditure of the system. However, the communication-awareness is severely limited and only used for the robot to position itself to enhance the link quality once the sensor has awaken. We shall advance beyond this simplified case to consider i) cooperation between adjacent sensor nodes using network coding, and ii) trajectory planning based on communication objectives and channel restrictions.

Network coding first appeared as an information theoretic multicast problem [1]. Refs. [14] and [15] showed linear codes over a network to be sufficient to establish any feasible multicast connection. The emergence of Random Linear Network Coding (RLNC) [16] led to practical applications in which the nodes in the network generate linear combinations of information symbols using random coefficients in a Galois field. Typically, these random coefficients are sent in the header of the packet that carries the coded symbols, which enables the receiver to learn the coding matrix and recover the original data by Gaussian elimination. Network coding is thus making a transition from an information theoretic result to real-life systems with applications to increased throughput [17], reduced delay [18] [19] [20], reduced energy [21], increased data persistence [2], enhanced peer-to-peer content delivery [22], data dissemination [23], and distributed storage [3][4]. Our work differs from previous work in that the dissemination and storage of the data is not driven only by the nature of the content, but also by the crucial interaction with the AV responsible for the final gathering of the data. A holistic, application-driven approach to dissemination, storage and data gathering using AVs has not been considered in the literature before our work.

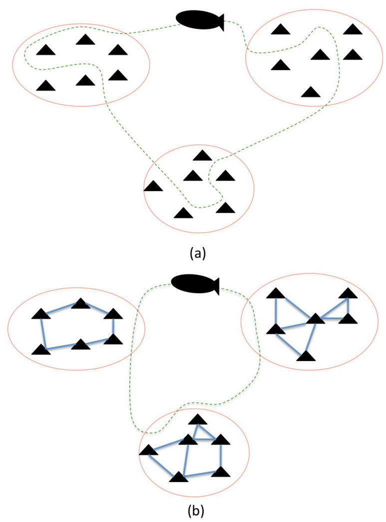


Fig. 1. Path selected for data gathering (a) without intra-cluster cooperation, and (b) with intra-cluster cooperation

III. Problem Definition

We focus on determining maximal paths to collect data from N geographically deployed sensors. The region is bounded and denoted by $X \subset R^2$. Each sensor node is denoted as S_i and its location is represented as $L_i = \{x_i, y_i\}, i = 1, \ldots, N$. The set of sensor nodes is represented as $\mathcal{N}$. The communication range of the sensor nodes could be large enough to allow communication with other sensors. Assume that there are K clusters and each cluster $C_k, k = 1, \ldots, K$ each consisting of a subset of connected nodes. For simplicity, any sensor node S_i is denoted as s.

We assume that there is a single AV, whose objective is to gather the data at the sensors and deliver it to a base station (BS) located at $L_b = \{x_b, y_b\}$. The AV location is denoted as $\bar{x} = \{x_a, y_a\}$ and has a velocity of $v\ m/s$. We assume without loss of generality that the AV is launched from the BS and must return to it. All sensors are capable of communicating to the AV and to each other if they are within communication range. We assume that the time for the AV to prepare and switch communication from one sensor to the another is negligible. We also assume that the AV has an omni-directional antenna. Each sensor s has M_s data packets to deliver to the BS, each of size b bits and assume that this information is known a priori to the AV. This assumption is valid for certain applications and deployments. For example, if a sensor deployment is measuring environmental data at a fixed sampling rate, the starting time of the sampling is known, and the starting time of the mission is also known, then a planner can have a good estimate of the data available in each sensor before starting the mission. Consider also the case of a system designed to gather the data of a certain time period, thus the amount of data available at each sensor is known at the time of planning.

For the present analysis, we focus on two extreme cases: i) no cooperation, where the AV needs to communicate to each and every sensor (Figure. 1 (a)), and ii) full cooperation, where the AV can communicate to a subset of the sensors in each cluster to retrieve the data because the sensors have previously exchanged it amongst each other (Figure. 1 (b)). For the latter, each sensor may contain more than its own data after the cooperation process and before the data is gathered by the AV. We emphasize that we do not impose the need of a cluster head that gathers the data. Also note that Figure. 1 provides a simple example to illustrate the benefits of cooperation towards reducing the path length to gather the data. Networking mechanisms and network coding play a role in the cooperation and data storage that precedes the AV's visit to different clusters. Finally, note that the first case can be though of as a special case of the second case where there is a single sensor node in each cluster.

For this work, we assume that the cost of cooperation for the sensors or, more specifically, the cost of data dissemination in preparation for the data collection process is negligible. Future work shall consider explicit trade-offs when the cost of cooperation is accounted. However, we do propose mechanisms to determine the minimum amount of data required at each node in order to provide optimal performance in terms of route planning.

We shall define $R_s(\bar{x})$ as the communication rate (goodput) of sensor s when the AV is at location $\bar{x}$. For case i) this metric constitutes the key to establishing communication to each sensor. When $\bar{x}$ is beyond R_{c_i} of sensor node s then $R_s(\bar{x}) = 0$. For case ii), the metric of interest for cluster k is $R^k(\bar{x}) = \max_{s \in C_k} R_s(\bar{x})$, where we assume that only one node can communicate with the AV at each time but that several nodes in cluster k can communicate to the AV during the mission to forward the clusters data. Communication is established through the cluster's sensor with the best goodput to the AV. Although we focus on two dimensions, a similar analysis can be carried out for three dimensions.

Note that $R_s(\bar{x})$ can incorporate a variety of effects, including modulation, physical (PHY) layer coding, antenna characteristics of sensor s, and channel characteristics. The simplest example would be considering an additive white Gaussian noise channel, where $R_s(\bar{x}) = B \log_2 \left(1 + \frac{P_{tx}}{N_0 B} \frac{1}{r_{s,x}^{\alpha}}\right)$, where P_{tx} is the transmission power, α is the path loss exponent, B is the transmission bandwidth, $r_{s,x}$ is the Euclidean distance between $\bar{x}$ and L_s, and N_0 is the noise spectral density.

We define the path of the AV for the overall data exchange process as $\gamma : I \rightarrow X$, where I is the interval over which the path is defined [33]. The path γ is a continuous function. Assume that as the AV moves along γ, it can communicate with a node s for time a_s to b_s. We define $\gamma_s = \{\bar{\delta}_s(t) : a_s \leq t < b_s\}$ as the communication path with sensor s, with starting time a_s and ending time b_s, and where $\bar{\delta}_s(t) = \{x_a(t), y_a(t)\}$ represents the position of the AV at each instant t.

A. Minimizing the Mission Time

If the goal is to minimize the mission time, i.e., the time it takes for the AV to return to the base station after collecting data from the sensor network, we focus on the amount of data present in each sensor that needs to be gathered by the AV. Let us define $\bar{M}_s$ as the amount of data in bits available at sensor s, but not necessarily generated solely in s. For the case of no cooperation, the amount of data in bits that can be gathered from each node s is given by

$$\bar{M}_s = M_s \cdot b \geq \int_{a_s}^{b_s} R_s(\bar{\delta}_s(t))|\bar{\delta}'_s(t)|dt, \tag{1}$$

where $|\bar{\delta}'_s(t)|$ represents the Euclidean norm of the derivative of $\delta_s(t)$.

For the case of cooperation, the constraints are different, because it is only needed for the cluster to deliver enough data. For each cluster C_k, we need

$$\bar{M}_{C_k} = b \cdot \sum_{s \in C_k} M_s \geq \sum_{s \in C_k} \int_{a_s}^{b_s} R_s(\bar{\delta}_s(t))|\bar{\delta}'_s(t)|dt. \tag{2}$$

However, each node will have a constraint related to how much data it contains (including its own data and that of others), i.e.,

$$(\bar{M}_s + r_s) \cdot b \geq \int_{a_s}^{b_s} R_s(\bar{\delta}_s(t))|\bar{\delta}'_s(t)|dt \tag{3}$$

where $r_s \geq 0$ constitutes the additional data from other nodes. The latter is valid if we assume the additional data stored at each sensor constitutes a random linear combination of packets of the entire cluster. Otherwise, if each sensor were to store data packets of other sensors, additional constraints would be necessary to account for this fact in order to guarantee that no data repetition occurs during the data gathering process and that the same data packet is not counted as separate data packets. Of course, if every sensor were to store a redundancy equivalent to all data packets of the cluster, network coding would not be necessary for the purpose of efficient data storage. However, it can still have impact on the data dissemination process in preparation for data collection, e.g., consider the dissemination problems in [23], [24], [25], [26].

Clearly, a joint optimization of the route of the AV and the data dissemination process must be performed. However, we show that we can decouple both problems (Theorem 1) under certain conditions, which allows us to find a data dissemination and storage distribution across the network that still guarantees optimal performance (Theorem 2).

Theorem 1: If $r_s = \sum_{t \in C_k, s \neq t} \bar{M}_t, \forall C_k, s \in C_k$, then the problem of determining optimal data collection route of an AV with cost-free cooperation is decoupled from the data dissemination problem.

Proof: The condition implies that all nodes in each cluster have exchanged their data with all other nodes in their cluster prior to the arrival of the AV. Any node of a cluster can provide all data. Solving the optimal route problem does not depend on the specific distribution of the data in the network. ∎

Theorem 2: If the optimal data collection route problem is solved under the assumption that $r_s = \sum_{t \in C_k, s \neq t} \bar{M}_t, \forall C_k, s \in C_k$, then the additional data, r_s^*, that need be present at each node besides its own is given by

$$r_s^* = \max\Big\{0, \int_{a_s}^{b_s} R_s(\bar{\delta}_s(t))|\bar{\delta}_s'(t)|dt - \bar{M}_s \cdot b\Big\}. \quad (4)$$

Proof: The proof follows from using Theorem 1 and realizing that the condition on r_s relaxes the problem of data distribution across each cluster. Once the path has been planned, the AV will communicate with a subset of the nodes to exchange data. Since each node can always communicate its own data, r_s^* provides a measure of how many additional independent linear combinations (independent from the ones collected in other nodes in the cluster) are necessary from each node s. An $r_s^* = 0$ may mean that either no communication with s is established or s communicates a fraction of its own data. ∎

Efficient data dissemination strategies can then be derived to provide the required r_s^*. Although we provide a connection of this problem with an information flow formulation that account with the network's connectivity graph to provide a lower bound on the cost of dissemination, specific strategies for that dissemination shall be the focus of future work.

B. Maximizing collected data under a fixed mission time

Most AVs (unmanned aerial, underwater or ground) have a maximum mission time determined by the amount of fuel that the vehicle can carry. Given a maximum mission time T for these scenarios, the goal is to determine a path γ that maximizes the data obtained from the sensor nodes under cooperative and non-cooperative frameworks. The path should begin and end at the base station and take less than T units of time. The objective of the problem for the non-cooperative sensor networks is:

$$\max \sum_{k\in\{1,\dots,K\}} \sum_{s\in C_k} \int_{a_s}^{b_s} R_s(\bar{\delta}_s(t))|\bar{\delta}_s'(t)|dt \quad (5)$$

$$\text{Subject to}: f(\gamma) \leq T \quad (6)$$

$$Eq.\ (1) \quad (7)$$

where $f(\gamma)$ is the time taken by the AV to travel the path γ.

Remark 1: For the case of cooperation between nodes, a similar optimization is derived where Eq. (1) is substituted by Eq. (3) and Eq. (2).

To maximize the objective function, the data acquired by the path must be maximized. A straightforward solution is to convert the route planning problem into a traveling salesman problem (TSP). The TSP solution will allow the vehicle to visit every node and collect the information. However, when the number nodes is larger, the path given by the TSP may not satisfy the fuel time constraint. In that case, some of the nodes must be removed optimally and a new TSP must be solved. As the TSP is NP-Hard, removing the nodes and solving a new TSP introduces additional and non-negligible complexity to the problem. Moreover, our general formulation considers a varying transmission data rate depending on the position of the AV, which cannot be incorporated into a TSP trivially. The solution provided by TSP is thus far from optimal and further optimization of the solution would be required, e.g., in terms of defining the position of the cluster that the AV must visit, which will depend on the rate and the amount of data available at the cluster. Therefore, we use particle swarm optimization (PSO) technique to determine an optimized path that will provide close to optimal solutions to the route planning problem. PSO has shown promise in determining close to optimal solutions for several applications with large number of dimensions and variables [27][28]. Inspired by these applications of PSO, we have adopted PSO as a mechanism for route planning.

Remark 2: Straightforward extensions of Theorem 1 and Theorem 2 can be derived for this case. The key challenge is that we have no guarantee that all data shall be collected in the fixed mission time.

C. Intra-Cluster Cooperation: Data Dissemination in Preparation for Data Collection

Although a joint optimization of the AV path and the sensor cooperation process (dissemination and storage) is required to yield optimal solutions, we have focused on mechanisms that decouple the two problems and which are optimal under certain conditions. For these mechanisms, once the amount of data to be collected from each node is determined by the path planning algorithm, a calculation of the cost of data dissemination follows naturally. We characterize an information flow formulation that provides the minimum cost for the data dissemination process under the assumption that the communication topology in each cluster is known.

Remark 3: We emphasize that the current approach may be used in an iterative algorithm that takes into account the cost of cooperation as one of its optimization objectives. The input to the problem studied below will be the amount of data collected from each node at each iteration of the algorithm.

Clearly, the dissemination problem is constrained to each individual cluster. We assume that the underlying wireless network can communicate using a broadcast communication channel, e.g., radio. The communication network in each cluster is represented by an hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{A})$, which constitutes a generalization of a graph. In an hyperarc, $i \in \mathcal{N}$ can be associated to one or several hyperarcs of the form $(i, J) \in \mathcal{A}$. An hyperarc considers a connection between node i and a subset of nodes $J \subset \mathcal{N}$, which are in transmission range of node i. Each hyperarc has a cost associated to the amount of data it transmits, i.e., $f_{(i,J)}$. A valid assumption for our problem is that the cost, say in terms of energy, increases linearly with the amount of transmitted data, i.e., $f_{(i,J)} = a_{(i,J)} z_{(i,J)}$, where $z_{(i,J)}$ is the amount of data through (i, J) and $a_{(i,J)}$ is a constant. This assumption is valid in wireless networks where the transmission power is fixed and the energy consumed for transmission depends only on the time the hyperarc is used. If the transmission power is the same for all transmissions of node i, this means that $a_{(i,J)} = a_{(i,J')}, \forall (i, J), (i, J') \in \mathcal{A}$. If the transmission power is adapted so that the receivers achieve at least a minimum Signal-to-Noise Ratio (SNR), the constant $a_{(i,J)}$ will depend on the transmission distance to the node that is farthest away and the propagation loss for that channel. For example, if the distance between i and $j \in J$ is r_{ij}, the maximum distance for hyperarc (i, J) is $r_{(i,J)} = \max_{j\in J} r_{ij}$, and $a_{(i,J)} \propto r^{\alpha}_{(i,J)}$ for a channel with a pathloss exponent of α. More complex propagation models can be considered, e.g., [29].

For deriving a lower bound on the minimum cost, we shall assume that perfect coordination can occur amongst the sensors in the cluster to avoid packet collisions. Each sensor is a source of data and a possible destination of the flow from the data dissemination problem.

Once the optimal route for the AV and the amount of data that will be collected from each sensor has been determined, we can extend the networks' connectivity hypergraph to include these constraints. Namely, we create a virtual hypergraph $\mathcal{H}_v = (\mathcal{N}_v, \mathcal{A}_v)$, with a new virtual node v_n (representing the AV) and additional hyperarcs of the form $(i, \{v_n\})$, $\forall i \in \mathcal{N}$, i.e., hyperarcs that connect v_n to each physical node in the network. The capacity associated to hyperarc $(i, \{v_n\})$ is $C_{(i,\{v_n\})} = \int_{a_i}^{b_i} R_i(\bar{\delta}_i(t))|\bar{\delta}'_i(t)|dt$ while the cost associated to using it is zero. We assume that the AV does not collect redundant/duplicated data. The node v_n constitutes the virtual destination of all the traffic.

Finally, determining the cost of dissemination of the data to the different sensors is solved as a multi-commodity flow problem for the virtual hypergraph. Assuming that the cost of transmission is separable and linear, the optimization problem can be expressed as follows:

$$\begin{aligned}
&\min \sum_{(i,J)\in\mathcal{A}_v} a_{(i,J)} z_{iJ} \\
&\text{subject to } z \in Z \\
&z_{(i,\{v_n\})} \le C_{(i,\{v_n\})}, \forall i \in \mathcal{N} \\
&z_{iJ} \ge \sum_{s\in\mathcal{N}_v} \sum_{j\in J} x^{(s)}_{iJj}, \forall (i,J) \in \mathcal{A}_v \\
&\sum_{\{J|(i,J)\in\mathcal{A}_v\}} \sum_{j\in J} x^{(s)}_{iJj} - \sum_{\{j|(j,I)\in\mathcal{A}_v\}, i\in I} x^{(s)}_{jIi} = \delta^{(s)}_i \\
&x^{(s)}_{iJj} \ge 0, \forall (i,J) \in \mathcal{A}_v, j \in J, s \in \mathcal{N}_v
\end{aligned} \tag{8}$$

with

$$\delta^{(s)}_i = \begin{cases} \bar{M}_s & \text{if } i = s, \\ -\bar{M}_s & \text{if } i = v_n, \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $x^{(s)}_{iJj}$ represents the flow associated with sensor s, sent through hyperarc (i, J) and received by nodes $j \in J$. Z represents the set of constraints on the amount of data passing through each hyperarc. Our formulation is inspired by [30].

IV. Path optimization of non-cooperative nodes

The path γ is typically a differentiable function. Determining a route is difficult due to the complexity of the problem. Hence, we discretize the path into a set of way-points and optimize the locations of these way-points such that it satisfies the mission time constraint and maximizes the objective function. While traveling from one way-point to another, the AV collects the data as described in Section III. The way-points are denoted as $W_m, m = 1, \dots, M$ and the path connecting the way-points constitutes the route for the AV. The PSO uses these way-points to optimize their location to maximize the data gain. A scenario with a discretized path is shown in Figure 2. The path in Figure 2 is defined through six waypoints W_1 to W_6. Note that the waypoints do not pass through the node centers Y_1 to Y_6. We describe the PSO basics and then provide a detailed account of how to obtain the solution.

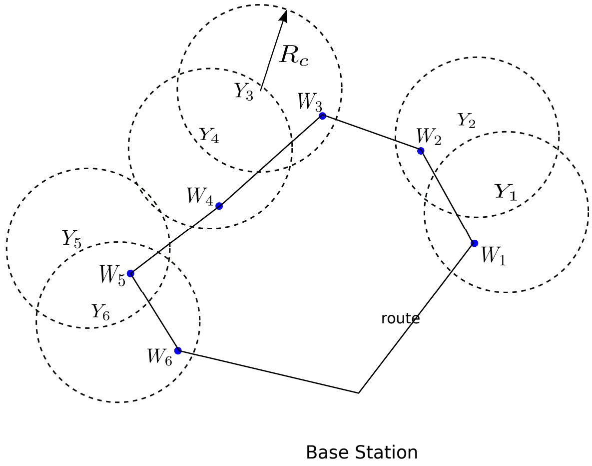


Fig. 2. Path of a potential PSO solution with the location of various way-points

A. Particle swarm optimization basics

PSO is a population based iterative optimization technique developed by Eberhart and Kennedy [31]. The key idea behind the algorithm is to simulate the social behavior of bird flocks, fish schools, etc. Each particle in a swarm is a potential solution in the search space. The particle adjusts its velocity according to its own flying experiences and its flock's experiences. The PSO technique is similar to the evolutionary computation techniques in [32], however, in PSO, each particle can adapt its velocity to move in the search space and has memory of its best position.

If we assume the optimization problem to be of Q-dimension, then each particle in the swarm S can be represented as $X_l = (x_{l1}, \dots, x_{lQ}), l = 1, \dots, |S|$, where $|S|$ represents the number of particles in the swarm. The best previous position attained by the particle is represented as $P_l = (p_{l1}, \dots, p_{lQ})$, and the velocity of the particle is $V_l = (v_{l1}, \dots, v_{lQ})$. The best global position achieved by the swarm is represented as $P_g = (p_{g1}, \dots, p_{gQ})$, and the iteration number is represented as superscript β. The particles in the swarm are updated according to the following equations

$$v^{\beta+1}_{ld} = \chi(wv^{\beta}_{ld} + c_1r_1(p^{\beta}_{ld} - x^{\beta}_{ld}) + c_2r_2(p^{\beta}_{gd} - x^{\beta}_{ld})) \tag{10}$$

$$x^{\beta+1}_{ld} = x^{\beta}_{ld} + v^{\beta+1}_{ld} \tag{11}$$

where $d = 1, \dots, Q$, r_1 and r_2 are uniformly distributed random numbers between $[0,\ 1]$, c_1 and c_2 are positive constants representing cognitive and social parameters, w is the inertia weight, and χ is the constriction factor. The role of inertia weight w is to create a balance between global and local explorations. The first iterations are targeted at exploring the search space with a large w is reduced as the solution approaches the optimal value [32]. The constants c_1 and c_2 aid in convergence of the solution. The random parameters r_1 and r_2 are used to maintain the diversity of the swarm population, while the constriction factor controls the effect of velocity on the particles. The process of evaluating a particle, updating the velocities and position of the particles, is carried out for Γ number of iterations.

B. Objective function for PSO

The dimension of the PSO depends on the number of way-points assumed for the solution. A smaller number of particles may not completely explore the search space X, while increasing the number of way-points results in a smoother path but at the cost of additional computational time. Hence, the number of way-points is a design parameter. Assume that the solution is designed using M way-points, and each way-point is denoted as $W_m = \{x_m, y_m\}, m = 1, \ldots, M$. Each potential solution of the PSO (say particle X_l) represents these M waypoints in a sequence, $W_1, \ldots, W_M$. Therefore, the dimension of the particle is $2M$ and $X_l = x_{l1}, \ldots, x_{l2M}$.

The cost associated with each particle X_l denotes two quantities – the mission time of the path ($\hat{T}_l$) and the associated data collected while traveling the path (J_l). If $\hat{T}_l < T$, only then J_l is evaluated, otherwise $J_l = 0$. The mission time of a particle (X_l) is determined as

$$\begin{aligned} \hat{T} &= \frac{D_t}{v} + \frac{D_b}{v}, \text{ where} \qquad (12) \\ D_t &= \sum_{\substack{d=1 \\ d=d+2}}^{2M-2} \sqrt{(x_{ld} - x_{l(d+2)})^2 + (x_{l(d+1)} - x_{l(d+3)})^2} \\ D_b &= \sqrt{(x_{l1} - x_b)^2 + (x_{l2} - y_b)^2} + \\ &\quad \sqrt{(x_{l(2M-1)} - x_b)^2 + (x_{l(2M)} - y_b)^2} \end{aligned}$$

where, $L_b = \{x_b, y_b\}$ is the base station location.

C. Initialization of PSO solution

The convergence of the PSO algorithm depends on how quickly PSO can determine an initial solution. As the search space for the route planning problem is large, determining a solution with random initialization can be time consuming. In order to speedup the process of finding a suitable solution, we initialize the PSO close to the location of the sensors. One such initialization can be through the TSP solution, but this requires the solution to have N waypoints. With large number of waypoints, the PSO dimension will increase significantly. For this reason, we initialize the PSO solution with M artificial nodes determined using k-means clustering. The k-means clustering determine k centers using the sensor node location information. For our problem, we seek $k = M$ centers. This type of initialization has two advantages. The first is that the center/artificial nodes will be in the vicinity of the sensor nodes, thus reducing the exploration for the PSO. The second advantage is that the dimension of the solution can be reduced to attain a good performance with low computational time.

To illustrate the difference in performance of the PSO solution using k-means initialization, we carried out a performance

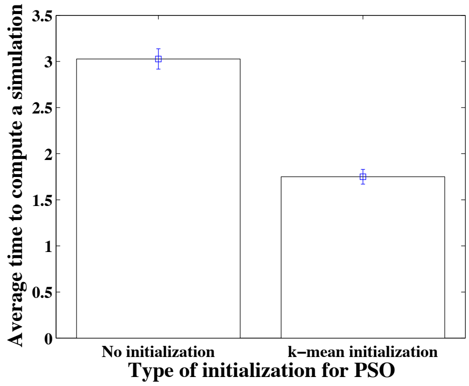


Fig. 3. Computation time of the PSO solution with k-means initialization and without initialization for a simulation with $M = 5$ and $N = 20$ to attain the same performance in terms of percentage of data packets. The simulation used the same PSO parameters for both the simulations.

evaluation for 20 sensors and $M = 5$ waypoints. The performance of the PSO solution with and without initialization is shown in Figure 3. k-means initialization enables the PSO to determine better solutions than without initialization. In the simulation section (Section VI), we use k-means initialization for all the simulations.

D. PSO solution

The route for the objective function is determined using the PSO update rules, objective function evaluation, and the initialization procedures described in Sections IV-A, IV-B, and IV-C. Algorithm 1 shows the process of determining the solution to the objective function defined in Equation 5. Initially, one of the particles is initialized using k-means clustering algorithm (Lines 2 and 3). Using the updated swarm, PSO routine is executed for a pre-determined number of iterations (Γ, Line 4).

In the PSO function, the time taken by each particle $\hat{T}$ is computed. If $\hat{T} \leq T$, then $J_l = 0$. Otherwise, the AV uses the route defined by the particle to collect data from the nodes. The data collected is accumulated into the vector J_l. The best particle with maximum cost is determined and the swarm positions and velocities are updated. This process continues for Γ iterations.

Note that the, initializing one of the swarm particles is useful only if $T > \bar{T}$, that is, the mission time T is larger than the time taken by the AV to travel along the route determined by the AV without stopping.

V. Path optimization of cooperative nodes

Sensors located in the communication range of each form cooperation clusters. In this case, the nodes can share data, which means that the AV need not travel to every node. Thus, it can visit fewer nodes collecting large amounts of data with smaller path length. Therefore, the route planning problem is to determine a path that will visit each cluster. A scenario is shown in Figure 1(b). In this case, we will leverage the algorithm used for the non-cooperative case, but modifying the way the objective function is evaluated by the PSO.

Algorithm 1 PSO solution for non-cooperative nodes

Init: $\mathcal{I}, \mathcal{N}, R_c, v, L_b$, Γ,Swarm[$|S|$][$2M$]
initialRoute = k-means algorithm(L_b, $\mathcal{N}$);
Swarm[1][1 to $2M$] $\leftarrow$ initalRoute
[route, costinfo] = PSO(Swarm, Γ)

PSO function

while $\Gamma > 0$ **do**
  for each $X_l, l \in S$ **do**
    Calculate $\hat{T}$ using Equation 12, and $J_l = 0$
    if $\hat{T} < T$ **then**
      while AV reaches base station **do**
        Find sensor nodes within communication range
        If sensor nodes are available then collect data from each node
        Update available information at the nodes
        Update AV position on the route
      end while
    end if
    $J_l \leftarrow datacollected$
  end for
  Determine the best particle of S and best cost
  Update the swarm positions and velocities using Equation 10 and 11.
  $\Gamma = \Gamma - 1$
end while
return(best particle, best cost)

A. Objective function for PSO

Each cluster C_k consists of a set of sensor nodes. Assume that the time taken for the data to arrive from one node to another in the cluster is negligible. Each cluster can be represented with a node having arbitrary communication range and shape. The range and shape is determined by the location of the sensor nodes in the cluster. A scenario of a cluster is shown in Figure 4. The validation of the path satisfying the mission time constraint is similar to that described in Section IV-B. The process of collecting data is given in the next section.

B. PSO solution for cooperative nodes

The process of determining a route that maximizes the data gained or collected per sortie with cooperative nodes is almost similar to that of the non-cooperative nodes. However, the amount data that a node can have is limited by $(\bar{M}_s + r_s) \cdot b$ in the cooperative sensor system, while the amount of data for non-cooperative sensor network is limited to $\bar{M}_s$. Therefore, the AV can collect more data by visit fewer nodes. We evaluate the performance in the next section.

VI. Simulations Results

We evaluate the performance of the path planning algorithms for cooperative and non-cooperative sensor networks (Sections IV-D and V-B) using Monte-Carlo simulations for different number of sensor nodes, mission time, and number of way-points. The performance of our PSO strategies is compared against three other algorithms, (i) sequential, (ii) greedy, and (iii) TSP-based data collection schemes. These schemes are intuitive schemes and provide a mechanism to compare our strategies. The greedy and sequential data collection strategies are simple to use, while the TSP approach provides a solution that optimizes the route traveled by the AV.

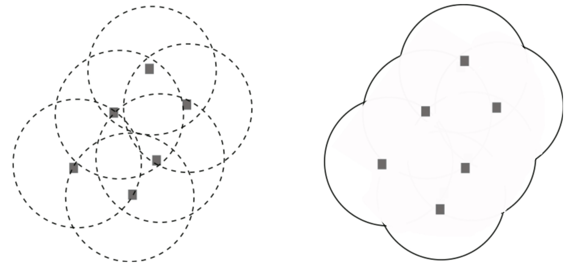


Fig. 4. A cluster with its communication ranges (left image) and the cluster with its total communication range and shape which is same for any node

A. Simulation setup

We carried out 250 simulations on an bounded region of $200m \times 200m$. The location of the sensor nodes are randomly generated using a normal distribution $\mathcal{N}(\mu, \sigma^2), \mu = 0, \sigma^2 = 1$. The number of sensors N deployed varies for different results, but we used in most cases $N = \{10, 20, 30, 40, 50, 75, 100\}$. We have evaluated the schemes for various mission times, namely $T = \{250, 500, 750\}$, to determine the effect T has over the average amount of data collected. The AV speed was fixed at $v = 2m/s$, which is a typical speed for underwater AVs. The maximum amount of data available for each node was randomly generated using normal distribution $\mathcal{N}(\mu, \sigma^2), \mu = 0, \sigma^2 = 1$, and the maximum amount of data that a node can have is limited to 50 units of data. In order to analyze the trade-off between accuracy and computation of the PSO solution, we also carry out simulations for different number of way-points $W = \{2, 4, 6, 8, 10\}$ and varying the number of sensors. The communication range of the sensor node is limited to $20m$

B. Performance of cooperative and non-cooperative solutions

We considered non-cooperative and cooperative sensor networks, a mission time of 500 time units, and a PSO solution having 8 way-points. The performance in terms of the average percentage of collected data for 250 simulations is shown in Figure 5. As expected, sensor cooperation allows the AV to collect a larger amount of data than in the case of non-cooperative sensor networks.

When the number of sensors is 10, the performance of non-cooperative and cooperative strategies is similar due to the limited number of connected sensors and thus the large number of clusters with very few sensors associated to each of them. When the number of sensors increases, the number of clusters decreases while the size of these clusters increases. Thus, the AV can collect a large amount of data while visiting fewer sensors.

Due to cooperation of nodes, the performance of the cooperative sensor network will always be higher than that achieved using non-cooperative sensor network. We will now analyze the effect of various parameters on these two extreme scenarios.

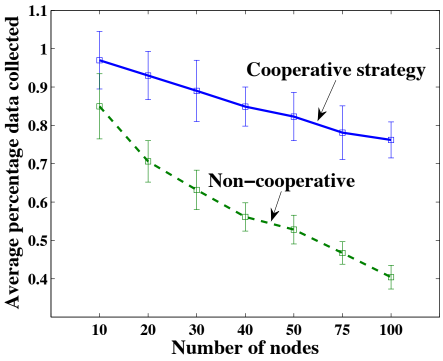


Fig. 5. Average percentage of collected data packets of the cooperative and non-cooperative strategies for different number of sensor nodes. The mission time is 500 time units and the number of way-points is 8

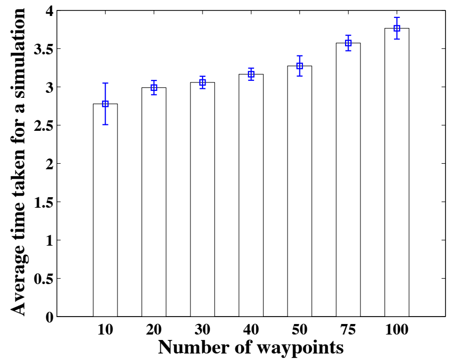
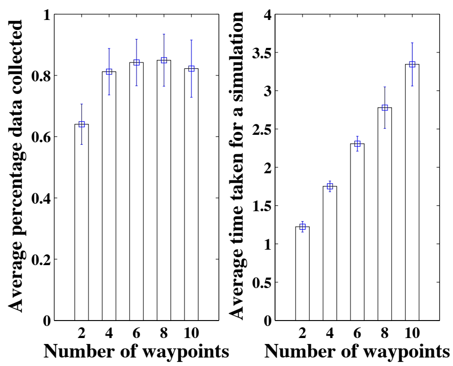
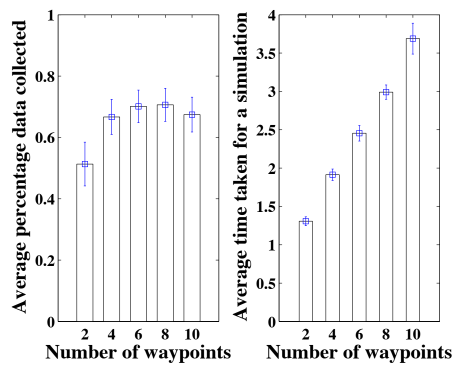


Fig. 7. The time required to carry out a simulation with different number of sensor nodes



(a)



(b)

Fig. 6. Average percentage of collected data packets and average computation time of the non-cooperative strategy for a mission time is 500 time units with different number of waypoints. (a) Number of nodes is 10. (b) Number of nodes is 20

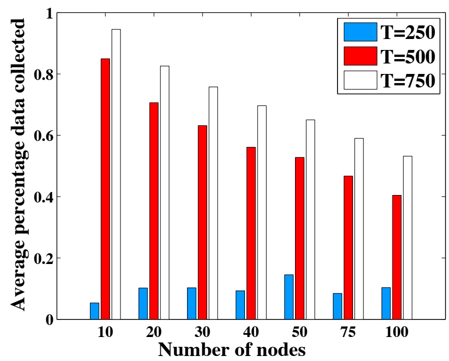


Fig. 8. Average percentage of data collected for different mission times (250,500 and 750) and different number of nodes

1) Effect of change in number of way-points: One of the main design parameter for PSO solution is the selection of number of way-points, M. While maintaining a fixed mission time and number of deployed sensors fixed, we study performance and computation time for various values of M. For our simulations, we fixed the number of iterations of the PSO algorithm to 4000.

The performance of the PSO solution in terms of average percentage data collected and the average time taken to perform a simulation along with their standard deviation bars is shown in Figures 6(a) and 6(b). Figure 6(a) shows that increasing the number of waypoints up to $M = 8$ improves the quality of the solution. However, a further increase in the number of waypoints degrades the quality of the solution due to the fact that the dimension of the search space increases significantly with a higher number of way-points and a larger number of iterations are needed to converge to a solution, i.e., more than the 4000 iterations considered in our approach. Figures 6(a) and 6(b) show that the computation time to carry out a simulation increases with the number of waypoints. Although the computation time is increasing, the average time taken for one solution of PSO with 10 and 20 sensors is close to 3.4 and 3.7 seconds, respectively, which is moderate. Another conclusion is that the PSO solution scales well with an increase in the number of sensors, with only small additional computation time when the number of sensors is doubled.

2) Scalability of PSO solution: In order to validate that our proposed PSO algorithms are scalable, we carried out simulations for a fixed number of waypoints and a mission time of 500 time units. Based on the results from Section VI-B1, we selected $M = 8$ waypoints for the simulations. The average time taken to execute a simulation with its associated standard deviation is shown in Figure 7. Note that the average time taken for a simulation consisting of 100 nodes is less than four seconds and that the increase in simulation time is quite mild, which shows that the proposed PSO solution is scalable for deployments with a large number of sensors.

3) Effect of mission time: The performance of the solution depends on the mission time. If the mission time is large while holding the number of waypoints and sensor nodes constant, then the AV can visit larger number of nodes collecting the data. However, when the mission time is smaller, it needs to optimize the solution in a better fashion to maximize the data collected. To analyze this situation, we carried out Monte-Carlo simulations for different mission time $T = \{250, 500$ and $750\}$. For a given mission time, the simulations were carried for varying number of nodes to ensure that the results are consistent. For all the simulations, the number of way-points is eight. The simulation results are shown in Figure 8.

From Figure 8, we can see that, for a given number of nodes, the performance of the non-cooperative strategy improves with increase in mission time. When number of nodes is 10, we can see that the percentage of data obtained with $T = 250$ is very low, while the data collected with $T = 750$ is high. The same type of pattern can be seen in the figure for simulations with other set of sensor nodes.

C. Comparison strategies

To compare the results of the cooperative and non-cooperative strategies, we now describe the sequential, greedy, and TSP based data collection schemes in detail.

1) Sequential data collection scheme: In this approach, the AV determines the nearest sensor node to it in terms of distance and data available to transfer. If the nearest node does not have any data to transfer, then that node is not considered during selection. After selecting a node, it moves towards the center of the node. As the AV travels along the route, it communicates with the nodes and collects data. It updates the amount of data available at each node after receiving data from the node. In this fashion, the AV continuously updates the data available at each node and also its position traveling towards the selected node center. While updating the AV position, the AV checks if it can reach BS within T from the current location L_i. If it can reach it, then it continues the mission, otherwise it will move towards the BS. The algorithm for this approach is given in Algorithm 2

Algorithm 2 Sequential data collection scheme

$\mathcal{N} = i, \ldots, M, = 0.v$
Determine the closest node $\hat{S}$ that has data and move towards $L_{\hat{S}}$
while $t + \frac{||\bar{x} - L_b||}{v} < T$ **do**
  Collect data from all the nodes AV can communicate
  Update t and available data from all the sensor nodes
  If reached $L_{\hat{S}}$ or $\bar{M}_{\hat{S}} = \emptyset$ then select $\mathcal{N} \setminus \hat{S}$, and determine the next $\hat{S} \in \mathcal{N}$
  If $\mathcal{N} == \emptyset$, then Break
end while

2) Greedy data collection scheme: In this scheme, the process of collecting the data within a sensor node is similar to the sequential data collection scheme. However, the sensor nodes are selected based on the amount of data they have. The algorithm for this approach is similar to Algorithm 2, except that $\hat{S}$ is selected as:

$$\hat{S} = \max_{i \in \mathcal{N}} \bar{M}_i.$$

3) TSP solution : In this approach, a path is generated using a TSP algorithm. The TSP algorithm determines a route through all the sensor nodes. Assume that the time taken by the AV to travel the route is $\hat{T}$. For a given fixed mission time of T time units, there is no guarantee that $\hat{T} \le T$ for all kinds of mission. In that case, the TSP solution is not valid. In the case of PSO solution, the PSO algorithm optimizes the way-point locations to maximize data collection. Similarly, to maximize the data collection for a TSP, we remove the minimum data contributing node from the list of sensor nodes. With the new set of nodes ($\mathcal{N}'$), we determine the TSP route time ($\hat{T}'$). If $\hat{T}' \le T$, then the new route is the solution of the TSP. Otherwise, the process is repeated recursively until a route is determined. In this fashion, the final modified TSP solution will visit those nodes that have high data capacity and satisfy the mission time constraint.

Note that in greedy and sequential strategies a node that has been visited can be re-visited if that node has data. However, in the case of TSP route, the AV cannot revisit the same node again. The route time of the TSP with the modified number of nodes can be significantly less than the mission time. To provide a fair comparison with other strategies, we allow the AV to trace back the visited route until time T' and return to the BS. Note that during the first pass of the route, if some nodes do not have data, then these nodes are removed and a new TSP solution is generated. The AV uses the new TSP route during the second pass.

D. Comparison results

The performance of the three comparison strategies against the PSO solution for non-cooperative and cooperative sensor network is studied through simulations considering 8 way-points.

Figure 9 shows the average percentage data collected by each strategy. From the figure, it is evident that the cooperative scheme performs far better than the non-cooperative and comparison strategies. An interesting result is that the TSP performs similar to that of the non-cooperative strategy for 10 sensors. This can be explained by the fact that the TSP route is able to visit most of the nodes when the number of sensors is small, thus collecting information similar to the PSO. However, with the increase in the number of nodes, the TSP solution is unable to interact with a higher number of nodes compared to the non-cooperative strategy degrading its performance with respect to the non-cooperative strategy, although still outperforming the more naive strategies (greedy and sequential).

Figure 9 shows that a deployment with 100 sensors can increase the amount of data collected by a factor of 3 with respect to the TSP-based approach and by a factor of 2 with respect to the PSO solution without cooperation.

From the results presented in this section, the proposed PSO solution for non-cooperative and cooperative strategies can provide quality solution for the joint optimization problem.

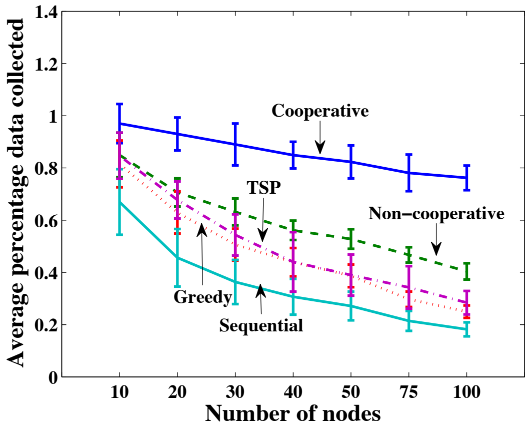


Fig. 9. Average percentage data collected for a mission of 500 time units with different number sensor nodes.

The proposed solution is scalable and multi-objectives can be handled easily. Finally, we observe that the performance gap between cooperative schemes and non-cooperative schemes increases with the number of nodes.

E. Discussions

The solution proposed in this paper can be used in several large scale data collection applications. For instance, the solution can be used for an AV to fetch data from sensors located at geographically distant places. Such applications include, collecting data from volcanic region, national parks, ocean observatories, etc.

There are several directions in which the scope of the problem can be extended considering the geographical displacement of the sensor nodes. If the nodes are dispersed with significant distance, then multiple AVs can be used to collect data. In this case, the AVs have to cooperate with each other to optimally allocate the nodes. Deploying multiple AVs will further enhance the mission robustness. In the case of underwater deployments, the presence of a low rate and high latency communication channel makes developing cooperative mechanisms a challenging task, but one that can significant impact the mission time.

VII. Conclusions

In this paper, we have analyzed fundamental characteristics and limitations of data gathering with AVs in sensor deployments organized in clusters of wirelessly connected sensors. For the case of intra-cluster cooperation, i.e., sensors within the cluster can disseminate their data to others, a joint optimization of the AV's route and the dissemination of data within each cluster is necessary. However, we proposed mechanisms to minimize the coupling between route planning and dissemination without loss in performance. The use of network coding reduces the required coordination amongst sensors and increases the ability of the system to gather large amounts of data compared to non-cooperative clusters.

We have also developed sub-optimal but efficient solutions for collecting large scale data from a set of sensor nodes. We proposed solutions for designing a path planner that maximizes the data gained from the sensor nodes under non-cooperative and cooperative frameworks. The solution is based on particle swarm optimization, which optimizes the locations of the way-points for the AV to maximize the data collection. The performance of our PSO schemes are compared against sequential, greedy, and TSP-based solutions. The simulations show that our PSO-based solutions are far better than the comparison algorithms. In the future, we will investigate how multiple AVs can be deployed, how practical cooperation mechanisms can be designed for large scale data collection, and study the effect of packet losses and the benefits that can be reaped from network coding in these scenarios. Future work will also consider the case of data with different priorities. If there is a notion of critical data in each sensor cluster, the algorithms could follow various approaches to guarantee the recovery of the critical data, from off-loading the critical data first when each cluster is visited to give different weights to critical packets in the path planning process so that the AV has an incentive to visit each cluster for sufficient time to gather the critical data packets.

References

[1] R. Ahlswede, N. Cai, S. Y. Li, and R. Yeung, “Network information flow,” IEEE Trans. Inf. Theory, vol 46, no 4, pp. 12041216, Jul. 2000.
[2] D. Munaretto, J. Widmer, M. Rossi, and M. Zorzi, “Resilient coding algorithms for sensor network data persistence,” In Procs of the 5th European Conf. on Wireless sensor networks (EWSN), Springer-Verlag, pp. 156-170, Bologna, Italy, Jan. 2008
[3] D. Leong, A. G. Dimakis, and T. Ho, “Symmetric Allocations for Distributed Storage,” In Proc. IEEE Global Telecomm. Conf. (GLOBECOM), pp.1-6, Hawaii, USA, Dec. 2010
[4] D. Leong, A. G. Dimakis, and T. Ho, “Distributed Storage Allocation for High Reliability,” In Proc. IEEE Int. Conf. on Comm. (ICC), pp. 1-6, Cape Town, South Africa, May 2010
[5] J.N. Tsitsiklis, ‘Efficient algorithms for globally optimal trajectories,” IEEE Trans. Autom. Contr., vol. 40, no. 9, pp. 1528-1538, Sep. 1995
[6] J. J. Enright, K. Savla, E. Frazzoli, and F. Bullo, “Stochastic and Dynamic Routing Problems for Multiple UAVs,” AIAA Journal of Guidance, Control and Dynamics, Vol. 32, No. 4, pp. 1152-1166, 2009.
[7] S. Rathinam, and R. Sengupta, “Lower and Upper Bounds for a Multiple Depot UAV Routing Problem,” In Proc. 45th IEEE Conf. on Decision and Control, pp. 5287-5292, San Diego, USA, Dec. 2006
[8] M Lindhé, and K H Johansson, “Communication-Aware Trajectory Tracking,” IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 1519 - 1524, Pasadena, USA, May 2008.
[9] O. Tekdas, N. Karnad, and V. Isler, “Efficient Strategies for Collecting Data from Wireless Sensor Network Nodes using Mobile Robots,” In Proc. 14th Inter. Symp. on Robotics Research (ISRR), Lucerne, Switzerland, 2009.
[10] A. Ghaffarkhah, and Y. Mostofi, “Communication-aware navigation functions for cooperative target tracking,” In Proc. American Control Conf. (ACC), pp.1316- 1322, Missouri, USA, Jun. 2009
[11] V. T. Le, N. Bouraqadi, S. Stinckwich, V. Moraru, and A. Doniec, “Making networked robots connectivity-aware,” In Proc. IEEE Int. Conf. on Rob. and Auto. (ICRA'09), pp. 1835-1840, Kobe, Japan, May 2009
[12] T. Ueda, S. Tanaka, B. Komiyama, S. Roy, D. Saha, and S. Bandyopadhyay, “ACR: an adaptive communication-aware routing through maximally zone-disjoint shortest paths in ad hoc wireless networks with directional antenna,” Wireless Comm. and Mobile Comp., Vol. 6, no 2, pp. 191199, Mar. 2006
[13] A. Larsen, O. B. G. Madsen, and M. M. Solomon, “Classification Of Dynamic Vehicle Routing Systems,” In Dynamic Fleet Management (Springer US), pp. 19-40, 2007
[14] S. Y. R. Li, R. W. Yeung, and N. Cai, “Linear Network Coding,” IEEE Trans. Inf. Theory, vol. 49, pp. 371, Feb. 2003
[15] R. Koetter, and M. Médard, “An algebraic Approach to network coding,” IEEE/ACM Trans. Netw., vol. 11, no. 5, pp. 782-795, Oct. 2003
[16] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong,“A Random Linear Network Coding Approach to Multicast,” Trans. Inf. Theory, vol. 52, no. 10, pp. 4413-4430, Oct. 2006

[17] S. Katti, H. Rahul, W. Huss, D. Katabi, M. Médard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” in ACM SIGCOMM, Vol 36, no 4, pp. 243-254, Pisa, Italy, Sep. 2006

[18] A. Eryilmaz, A. Ozdaglar, and M. Médard, “On delay performance gains from network coding,” Proc. 40th annual Conf. on Inf. Scie. and Syst. (CISS), pp. 864–870, New Jersey, USA, Mar. 2006

[19] T. K. Dikaliotis, A. G. Dimakis, T. Ho, and M. Effros, “On the delay of network coding over line networks, in Proc. IEEE Int. Symp. Info. Theory (ISIT), Seoul, Korea, Jun. 2009

[20] J. Barros, R. A. Costa, D. Munaretto, and J. Widmer, “Effective delay control in online network coding,” in Proc. 28th IEEE Conf. on Comp. Comm. (INFOCOM), Rio de Janeiro, Brazil, Apr. 2009

[21] D. E. Lucani, M. Stojanovic, and M. Médard, “Random Linear Network Coding for Time Division Duplexing: Energy Analysis,” In Proc. IEEE Int. Comm. Conf. (ICC), Dresden, Germany, Jun. 2009

[22] C. Gkantsidis, and P. Rodriguez, “Network coding for large scale content distribution, in Proc. Joint Conf. of the IEEE Comp. and Comm. Societies (INFOCOM) , vol.4, pp. 2235- 2245, Miami, USA, Mar. 2005

[23] D. E. Lucani, M. Médard, M. Stojanovic, and D. R. Karger, “Sharing Information in Time-Division Duplexing Channels: A Network Coding Approach,” In Proc. 47th annual Allerton Conf. on Comm. Control and Comp., pp.1403-1410, Illinois, USA, Sept. 2009

[24] D. E. Lucani, F. H. P. Fitzek, M. Médard, and M. Stojanovic, “Network coding for data dissemination: it is not what you know, but what your neighbors don’t know,” In Proc. 7th Int. Conf. on Mod. and Opt. in Mob., Ad Hoc, and Wir. Net. - (RAWNET Workshop), pp. 407-414, Seoul, South Korea, Jun. 2009

[25] S. Deb, M. Médard, and C. Choute, “Algebraic gossip: a network coding approach to optimal multiple rumor mongering,” IEEE Trans. on Info. Theory , vol.52, no.6, pp. 2486- 2507, Jun. 2006

[26] B. Haeupler, “Analyzing network coding gossip made easy,” In Proc. 43rd annual ACM Symp. on Theory of Comp (STOC), pp. 293-302, San Jose, CA, USA, Jun. 2011

[27] K. Parsopoulos, and M. Vrahatis,“Particle swarm optimization method in multiobjective problems,” In Proc. of the ACM Symp. on Applied Comp. (SAC), pp. 603–607, Madrid, Spain, Mar. 2002

[28] C C. Coello, and M. Salazar Lechuga, “MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization,” In Proc. IEEE Cong. on Evol. Comp. (CEC). pp. 1051–1056, Hawaii, US, May 2002

[29] D. E. Lucani, M. Médard, and M. Stojanovic, “Underwater Acoustic Networks: Channel Models and Network Coding Based Lower Bound to Transmission Power for Multicast,” IEEE J. Sel. Areas Commun., vol.26, no.9, pp.1708-1719, Dec. 2008

[30] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, “Minimum-Cost Multicast Over Coded Packet Networks”, IEEE Trans. Inf. Theory, vol. 52, no. 6, pp.2608-2623, Jun. 2006

[31] R.C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” *Proc. of the Symposium on Micro Machine and Human Science*, Piscataway, NJ, pp. 3943, 1995

[32] K.E. Parsopoulos and M.N. Vrahatis,“Recent approaches to global optimization problems through particle swarm optimization,” *Natural Computing*, Springer, Vol. 1, pp. 235-306, 2002

[33] S.M. LaValle, “ Planning algorithms,” Cambridge University Press, 2006

P.B. Sujit is a research scientist in the Department of Electrical and Computer Engineering, Universidade do Porto, Portugal. He completed his bachelor’s degree from Bangalore University, Bangalore in 1998, Master’s degree from Visveswaraya Technological University, Belgaum, India in 2002, and PhD from the Indian Institute of Science, Bangalore in 2006. Before joining Universidade do Porto, he worked as a post-doctoral fellow at Brigham Young University between 2006-08. His research interests include unmanned aerial and underwater vehicles, multi-robot systems and human-robot interaction.

Daniel E. Lucani is an Assistant Professor at the Faculty of Engineering of the University of Porto and a member of the Instituto de Telecomunicações (IT). He received his B.S. (*summa cum laude*) and M.S. (with honors) degrees in Electronics Engineering from Universidad Simón Bolívar, Venezuela in 2005 and 2006, respectively, and the Ph.D. degree in Electrical Engineering from the Massachusetts Institute of Technology (MIT) in 2010. His research interests lie in the general areas of communications and networks, network coding, information theory and their applications to highly volatile wireless sensor networks, satellite and underwater networks, focusing on issues of robustness, reliability, delay, energy, and resource allocation. Prof. Lucani was a visiting professor at MIT. He is the general co-chair of the Network Coding Applications and Protocols Workshop (NC-Pro 2011) and has also served as reviewer for high impact international journals and conferences, such as, the IEEE Journal of Selected Areas in Communications, IEEE Transactions on Information Theory, IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Journal of Oceanic Engineering.

João Borges Sousa is with the Electrical and Computer Engineering Department from Porto University. He is also the head of the Underwater Systems and Technologies Laboratory, an interdisciplinary research laboratory established in 1997 with researchers drawn from Electrical, Computer, and Mechanical Engineering and from Computer Science. His research interests include unmanned vehicles, control architectures, control, and coordination. He has been leading the design, implementation and deployment of advanced unmanned air and ocean vehicle systems in projects funded by the Portuguese Science Foundation (FCT), the Portuguese Ministry of Defense, the Portuguese Innovation Agency (ADI), the European Union (FP6/FP7), NATO and, in the United States of America, the Office of Naval Research and DARPA. In 2006 the laboratory received the national BES Innovation National Award for the design of the Light Autonomous Underwater Vehicle. In 2007 he received an outstanding teaching award from Porto University and has authored over 220 scientific papers.