IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 11, 2021, accepted June 27, 2021, date of publication July 8, 2021, date of current version July 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095610

Optimal Configuration Planning for Sensor Network Serviceability Under a System Coverage Constraint

THOMAS A. WETTERGREN, (Senior Member, IEEE), AND RUSSELL COSTA, (Senior Member, IEEE)
Naval Undersea Warfare Center, Newport, RI 02841, USA

Corresponding author: Thomas A. Wettergren (thomas.wettergren@navy.mil)

This work was supported by the Office of Naval Research under Award N0001420WX00258.

ABSTRACT We develop an optimization approach for the planning problem of configuring the positions of sensors within a sensor network for minimization of the travel length required to service the sensor locations. This is in contrast to existing approaches whereby the coverage of the sensor network is an objective of the optimization; in this new optimization approach the level of coverage is treated as a constraint. By beginning with an over-populated sensor network and then alternating between sensor repositioning and sensor removal, we create an optimization procedure that solves this difficult practical planning problem. A formal rule for switching between the repositioning and removal components of the optimization strategy is developed. Numerical example problems are presented to illustrate the method and show its performance against a heuristic optimization approach.

INDEX TERMS Genetic algorithms, optimal planning, Q-coverage, traveling salesman problem, wireless sensor networks.

I. INTRODUCTION

The planning of the positions of sensors within a large-scale sensor network is a common problem in many areas of application, ranging from climate monitoring and wildlife tracking, to the surveillance of sensitive areas against intruders. Historically, the configuration of these positions has been determined by optimizing the coverage performance of the sensor network for some fixed number of available sensors. As the sensor technology has matured, the sensors are becoming more complex and, maintaining a sensor network over time can involve the need for a servicer to visit each of the sensor locations. In addition, when communications are difficult or expensive, a servicer may need to visit each sensor location as a ''data mule'' to collect information from the sensors. Such systems have been developed for applications in long-term underwater monitoring of coral reefs and fisheries [1] as well as land-based monitoring of agricultural fields [2]. In such cases, the serviceability of the sensor network, defined as the minimal tour length required to visit each of the sensor locations, is a critical design consideration that is overlooked in current sensor network configuration procedures. Even when not visiting for maintenance, oftentimes the initial deployment calls for careful visitation of each site, as many applications require precise positioning to meet performance goals. Such deployments have been performed with autonomous helicopters deploying ground sensors [3].

A unique aspect of this type of sensor positioning problem is that the optimization of the positions of sensors is performed not to optimize a network performance objective (such as coverage), but instead to optimize the serviceability of the network under a constraint (such as meeting a coverage demand). This leads to a different type of optimization problem that has not been previously examined, and a reliable optimization approach to achieve good solutions to such problems is required.

In this article, we solve this novel sensor placement problem, wherein the objective under consideration is the serviceability of the deployed sensors. We create a genetic algorithm based solution approach to positioning these sensors to optimize the serviceability under a coverage constraint. In addition, we consider the removal of any redundant sensors (sensors that are close enough to others that they do not contribute significantly to coverage yet their presence

The associate editor coordinating the review of this manuscript and approving it for publication was Peng-Yong Kong.

increases the serviceability) as an additional heuristic procedure that couples with the genetic algorithm. Two variations of this optimization approach are developed for solving this problem, and the efficacy of the approaches are compared relative to a direct heuristic sensor placement optimization approach on numerical examples.

II. BACKGROUND ON SENSOR NETWORK CONFIGURATION OPTIMIZATION

When sensors are placed for use in a sensor network, the resulting group of sensor positions corresponds to some spatial pattern or configuration. We thus refer to the optimization of these configurations to achieve some performance goal as the *sensor placement problem*. The sensor placement problem (or Q-coverage problem) is a probabilistic variant of the facility location planning problem [4]. Q-coverage is a sensor network performance metric in which specific locations must be covered by (potentially multiple) sensors to achieve a desired quality of service [5]. In particular, the sensor placement problem is a cooperative covering problem [6], [7], whereby the ''coverage'' of regions by a sensor is not a binary effect (i.e. covered or not-covered) but instead is represented by a probability of coverage that depends upon the probabilistic combination of all sensors within range of that region. Reference [8] provides an overview of the variety of coverage problem formulations that have been developed for sensor network placement problems.

While Q-coverage is important in sensor network planning for surveillance applications, there are other metrics that have been applied to sensor network placement in other application contexts. When visual sensor networks are employed, the planning optimization includes not only the location of the sensors but also their orientation (as the field of view is not omnidirectional). In many of these visual sensor networks, there is a goal to place the network in a configuration so as to optimize network lifetime [9], [10]. In other applications, network lifetime is treated as a constraint [11]. However, we posit that lifetime can also be improved in situations by explicitly considering the serviceability of the sensor nodes as an objective.

The general Q-coverage planning problem has been shown to be NP complete and, for that reason, a variety of metaheuristic optimization approaches have been examined for planning sensor placements. Reference [12] separates the existing types of placement optimization techniques into four categories: (i) genetic algorithms, (ii) computational geometry, (iii) artificial potential fields, and (iv) particle swarm optimization. Genetic algorithm approaches represent the largest number of the methods surveyed. Other metaheuristic approaches that have been used include a simulated annealing optimization approach to place sensors on a grid for a problem of minimizing target discrimination error [13]. Additionally, ant colony optimization has been used [14] to maintain both sensor connectivity and coverage while reducing the load level across individual sensors (to increase network lifetime).

Reference [15] developed a basic genetic algorithm for sensor placement and showed that it is efficient compared to greedy solution techniques. This result is similar to other studies on using genetic algorithm approaches for the related facility location planning problem [16]. These genetic algorithm approaches can function with improved efficiency by careful tuning of the genetic algorithm parameters. For instance, by examining the relationship between the sensor network phenotype representation and the associated genotypes, more efficient genetic algorithm approaches have been developed for sensor placement to optimize coverage [17]. Reference [18] shows various Q-coverage problems where they combine a genetic algorithm and integer linear program (ILP) to have the genetic algorithm first develop a ''master problem'' that is then solved by the ILP. This combined approach allows the genetic algorithm to get to a neighborhood of the global optimum and then the ILP performs the local optimization.

There are also a few specific sensor placement optimization problems that can be solved by linear programming methods without the need to initially run a metaheuristic. For example, an ILP approach was used to select a minimal set of sensors from a dense set of potential sensor locations on a grid [19]. In addition, the submodularity inherent in sensor placement problems (i.e. diminishing returns for adding additional sensors) allows a mixed integer programming (MIP) approach to achieve performance within known bounds of the true optimum [20]. Following this line of reasoning, reference [21] shows that a greedy placement of sensors can still provide near optimal positions even with non-uniform sensor range (i.e. not submodular) when there exists a dense enough coverage region. Other heuristic-style sensor placement approaches utilize a genetic algorithm to select a set of sensors from a dense set of potential sensor locations in order to achieve a coverage constraint with the smallest number of sensors [22]. Recent work includes using both coverage and connectivity as objectives, and these approaches either select sensor locations to activate [23] or else select specific sensors to deploy [24].

As with most practical engineering problems, the sensor placement problem often has multiple competing objectives. While most studies limit focus to one of the objectives, there have been some studies that examine the trade-off problem directly through Pareto optimization. Reference [25] surveyed various trade-offs that have been studied in the wireless sensor network optimization literature. The prior studies that were surveyed in that article look at a variety of objectives, however, none of those surveyed look at the objective that we refer to as *serviceability*. Much like the single objective approaches, the Pareto optimization approaches fall into broad categories of either local optimization methods or metaheuristics for approximate global optimization. For example, in one case a Proximity Avoidance Coverage-preserving Operator (PACO) was used to provide local optimization for a Pareto optimization problem [26]. That problem was to jointly minimize the number of nodes while also minimizing the energy utilization of the most-utilized node. For global optimization, differential evolution (DE) has been applied as an optimization approach to compute Pareto trade-off surfaces for sensor placements using coverage, connectivity and lifetime/energy [27]. In our prior work [28], we have used genetic algorithm approaches to determine sensor placement patterns that achieve a trade-off between coverage and false alarms.

Once the sensors are placed in a location to be utilized for an extended period of time, there is often a requirement to periodically visit each of the sensors. This can be done to serve many purposes: such as to extract the collected data, to perform limited basic maintenance, to provide battery charging, and/or to replace the sensors' batteries. When in a difficult environment, the use of a mobile unmanned agent to perform these visits is preferable, and therefore making the visits with a minimal amount of travel is beneficial. Reference [29] used a set of mobile robots to collect the data from the sensors and return to a centralized sink node. In a similar study [30], an approach was developed to use a mobile sink to visit each of the sensors to collect that data directly. In any such application, there is an additional design objective that the total path length for visitation of all of the sensors is to be minimized (or at least held below some design threshold).

For some data gathering applications, the visitation of neighborhoods of the sensors can be more efficient than visiting all of the sensors directly [31]. This type of tour planning problem has been referred to as the single-hop data gathering problem [32] wherein the NP-hard routing problem is solved heuristically. Other approaches for finding the optimal tour for visiting a fixed set of sensor locations include combining a genetic algorithm with a 2-Opt heuristic [33], applying artificial bee colony algorithms [34], and applying discrete firefly algorithms [35]. Yet another type of sensor network visitation problem involves routing to a fixed set of locations based on temporal demands from the sensors in periodic recharging; methods to solve such problems involve gravitational search [36] and PSO/GA hybrids [37]. All of these approaches involve visitation of a fixed set of locations, they do not consider adjusting the sensor placements (locations) to improve the visitation route.

There are multiple application areas in which the placement of sensors needs to be made with consideration of both the coverage and the cost of visiting the set of locations. In wildlife monitoring networks, data is often extracted physically from each of the sensors and therefore there is a requirement to visit each node periodically, yet the sensors must still be positioned to achieve the required coverage. Thus, in that example, positioning sensors to optimize visitation while maintaining a coverage constraint is an important system-level optimization problem. Another example is in extreme physical environments, such as underwater networks, periodic maintenance is a requirement and movement to visit spatially separated locations is costly and time-consuming. Thus, in that application the positioning of sensors to optimize visitation is an important goal (while also maintaining the coverage quality that the network was created for). Hence, the multiobjective problem of positioning to achieve both coverage and serviceability is a practical engineering design/deployment concern.

In this article we solve a multiobjective sensor placement problem where the objectives under consideration are the Q-coverage and the serviceability of the sensors. We handle the multiobjective nature of the problem by treating the Q-coverage part of the problem as a constraint. This is a practical consideration in that most sensor networks are deployed only when coverage can be obtained at a given minimal level. We then focus on optimizing the serviceability, which we define as the distance required for a servicing agent to visit each of the sensors in sequence. A further complication to our problem is that serviceability can often be improved by reducing the number of sensors included. Thus we optimize both the number and placement of sensors to achieve these goals. To our knowledge this is the first example of solving such a problem in sensor network configuration optimization.

III. PROBLEM FORMULATION

The problem to be solved is to optimize the placement of sensors (and also select the number of sensors to place) in order to minimize the travel distance to service the sensors, while providing the best coverage for the sensor network over the domain. These goals are typically in conflict, as providing extensive coverage typically requires a large number of sensors placed at spatially varying locations, whilst such a configuration requires a significant travel distance to visit each sensor location for service. On the other hand, a network of fewer sensors placed in either a single location or a simple configuration (such as a single ring) has much improved serviceability, albeit at a cost of decreased coverage. To handle this conflict, we formulate the problem as an optimization of the system's serviceability under a coverage constraint.

Consider the placement of m sensors $\{S_1, \ldots, S_m\}$ that are to be located in a closed planar domain $\mathcal{W} \subset \mathbb{R}^2$. Each sensor S_i is to be placed at a fixed location $x_i \in \mathcal{W}$. The sensors are omnidirectional and are all of identical performance, and we assume the coverage range $r_i = r_{cov}(x_i)$ of each sensor S_i depends only on its location x_i. Like most sensor placement studies, we assume the sensors to be omnidirectional; however, the effects of sensor directionality have been previously examined in other placement studies [38], its impact on this problem is a subject of future interest. The sensors perform coverage by observing within a disk of radius r_i centered at x_i with a probability p_d (where $0 < p_d \leq 1$). That is, the subset $\mathcal{W}_i$ of the domain $\mathcal{W}$ given by $\mathcal{W}_i(x_i) = \{x \in \mathcal{W} : ||x - x_i||^2 \leq r_i^2\}$ is provided service with a coverage quality of p_d. We use the service coverage quality $C(x)$ as a spatially-dependent measure of performance for any location $x \in \mathcal{W}$; for example, $C(x) = 0.9$ implies location x is covered with 90% effectiveness. In regions of overlapping coverage from multiple sensors, the performance of the sensor network within the domain is given as follows [12]:

$$C(x; x_1, \ldots, x_m) = 1 - \prod_{i=1}^{m}(1 - C_i(x; x_i)), \quad \forall x \in \mathcal{W} \quad (1)$$

where

$$C_i(x; x_i) = \begin{cases} p_d, & x \in \mathcal{W}_i(x_i) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

and the aggregate coverage of the entire domain is given by

$$C_{\mathcal{W}}(x_1, \ldots, x_m) = \frac{\int_{\mathcal{W}} C(x; x_1, \ldots, x_m)\, dx}{\int_{\mathcal{W}} \mathbf{1}\, dx}. \quad (3)$$

We note that the coverage formulation in equation (1) explicitly accounts for the effects of overlapping coverage. There is a performance requirement that, as a group, the sensors maintain a minimal quality of coverage C_{min}, which we refer to as the coverage demand. The number of sensors that are to be used is also bounded by the number of available sensors m_{max}, such that $m \leq m_{max}$.

In addition to meeting the coverage demand, there is a requirement to minimize the cost to service the sensors. This service cost is given as the minimum required travel distance to reach each of the sensors sequentially, coming from a defined entry point into the domain and exiting from that same point. We refer to this cost as the *service length* L. Thus, the service length L for a specific configuration of sensors $\{S_1, \ldots, S_m\}$ is given by the minimal tour length between the $m + 1$ locations given by the m sensor locations $\{x_1, \ldots, x_m\}$ plus an additional depot at location x_0. The minimum tour length that meets these requirements for a given configuration is a solution to the classic *traveling salesman problem* (TSP) [39].

The optimal TSP solution for the service length L associated with a configuration is specifically given by the solution to the following optimization problem:

$$L(x_0, x_1, \ldots, x_m) = \min \sum_{i=0}^{m} \sum_{j \neq i, j=0}^{m} d_{ij}(x_i, x_j) z_{ij} \quad (4)$$

$$\text{s.t. } 0 \leq z_{ij} \leq 1, \quad i, j = 0, \ldots, m \quad (5)$$

$$\sum_{i=0, i \neq j}^{m} z_{ij} = 1, \quad j = 0, \ldots, m \quad (6)$$

$$\sum_{j=0, j \neq i}^{m} z_{ij} = 1, \quad i = 0, \ldots, m \quad (7)$$

$$\sum_{i \in Z} \sum_{j \in Z} z_{ij} \leq |Z| - 1, \quad \forall Z \subseteq \{1, \ldots, m\} \quad (8)$$

where $d_{ij}(x_i, x_j)$ is the distance between sensors S_i and S_j (given by $d_{ij}(x_i, x_j) = ||x_i - x_j||$) and z_{ij} is an indicator variable that has a value of $z_{ij} = 1$ if the service tour includes a direct link between sensor S_i and sensor S_j, and $z_{ij} = 0$ otherwise. We utilize the Dantzig-Fulkerson-Johnson formulation [40] in the above, which is a well-known integer linear programming formulation of the classic TSP that includes a constraint (8) for subtour elimination. Another formulation that includes subtour elimination, the Miller-Tucker-Zemlin formulation [41], is more commonly used for TSP route optimization because it is often computationally preferable when applied to complete TSP problems. However, we have found that in this application, an iterative solution with increasing numbers of constraints of the above form provides a much faster solution. In particular, we apply an iterative relaxation of subtours [42], whereby only those subtours that appear in a solution are added in a constraint of the form (8) and the ILP is solved again, with the process repeating until no subtours exist. This iterative relaxation is very efficient in this application, as the increasing number of constraints that are added is typically orders of magnitude smaller than the initial consideration of all potential subtours. This approach still retains the full integer linear program nature of the problem and thus provides global solutions that can be found with a standard ILP solver.

The general sensor placement problem is thus to find an optimal configuration of sensors $\{S_1, \ldots, S_m\}$ (given by their locations $\{x_1, \ldots, x_m\}$) that solves the following optimization problem:

$$\{x_1, \ldots, x_m\} = \operatorname{argmin} L(x_0, x_1, \ldots, x_m) \quad \text{s.t. } C_{\mathcal{W}}(x_1, \ldots, x_m) \geq C_{min} \quad (9)$$

where $L(x_0, x_1, \ldots, x_m)$ is the service length given by the solution of the TSP problem in equations (4)-(8) and $C_{\mathcal{W}}(x_1, \ldots, x_m)$ is the coverage quality given by equation (3). The number of sensors m that are to be deployed in the sensor network is an additional parameter of the optimization, which is to be chosen under the simple constraint $m \leq m_{max}$. The selection of the number m is achieved jointly with the optimization problem given by equation (9).

IV. OPTIMIZATION APPROACH

The service length minimization problem posed in equation (9) has an objective function of which the numerical evaluation requires the solution of a secondary optimization problem (specifically, to determine the length of the service route for a set of locations requires the solution of a TSP optimization problem) with a constraint given by a nonlinear function representing the Q-coverage. Additionally, the problem has the added complexity of achieving this prescribed coverage demand with the smallest number of sensors possible. To handle the complexity of this optimization we propose a solution technique based upon a modified genetic algorithm. This approach alternates between using a genetic algorithm to find sensor placements that lower the service length L, and removing sensors that are redundant to meeting the coverage demand. This additional step of removing sensors provides a heuristic approach to meeting the coverage demand constraint with a smaller number of sensors, when possible, where the sensor removal process is separated from the sensor placement process.

To alternate between the two optimization procedures (placing available sensors and removing sensors that are redundant to coverage) we apply a stopping criteria to the genetic algorithm. The stopping criteria that we use is to stop the genetic algorithm procedure when its convergence slows significantly. To formalize evaluation of the slowdown in convergence, we utilize a measure of diversity of the population in the genetic algorithm and stop when it is degraded by a set amount. The combined optimization procedure for alternating between the genetic algorithm for sensor placement and the heuristic for removal of sensors is shown in flowchart form in Fig. 1, which we refer to as the *move-first* optimization procedure. This procedure assumes the genetic algorithm procedure is run first; obviously, one can alternatively begin with the removal heuristic and that leads to a slight variant of the combined optimization procedure as shown in flowchart form in Fig. 2, which we refer to as the *subsample-first* optimization procedure.

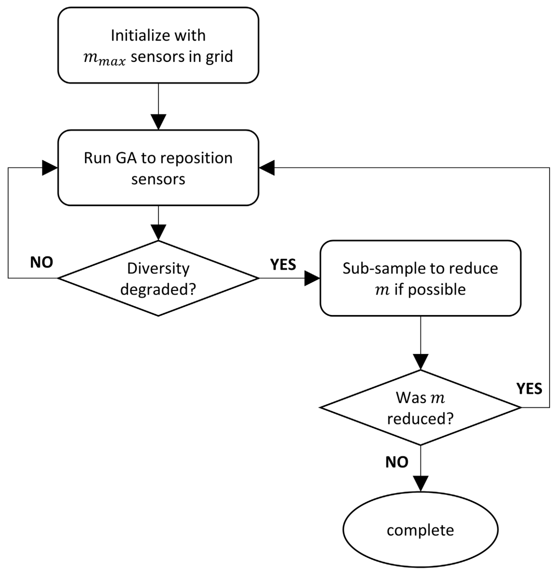

FIGURE 1. Flowchart of the move-first optimization procedure.

To maintain high quality optimization solutions from a genetic algorithm, the algorithm must maintain a level of diversity across the members of the population at each generation. The maintenance of diversity across the population of individuals as genetic algorithms progress has been recognized as important to convergence since early in their application [43]. If the diversity degrades too much, there will be too many replicates of similar individuals in a population, and that can lead to premature convergence. Usually, the analysis of a genetic algorithm uses a genotypic measure of diversity given by the Hamming distance between members of the population [44]. That measure is genotypic in that it measures the distance between individuals in the space of the genome representation (typically a binary string). Alternatively, it is possible to consider phenotypic measures of diversity that

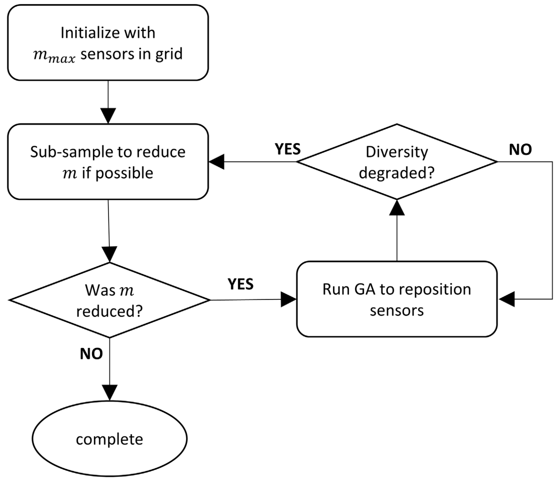

FIGURE 2. Flowchart of the subsample-first optimization procedure.

measure the distance is a more natural/physical interpretation of the distance between individuals. The method that we utilize for diversity measurement is a phenotypic measure called inertia [45]. We have previously had success in using this measure to examine genetic algorithm performance for the optimization of spatially distributed nodes on a grid [46].

The measurement of inertia for the group of individuals in a population for a specific generation of a genetic algorithm is related to the moment of inertia for mass distribution in high-dimensional spaces. Particularly, consider a set of P members in a population with γ characteristic traits. Let the value of the i-th trait of individual p be given by t_{ip}. Then the centroid of the value of the i-th trait is given by

$$c_i = \frac{1}{P}\sum_{p=1}^{P} t_{ip}, \tag{10}$$

and the inertia of the group of P individuals across all i traits is given by

$$I = \frac{1}{\gamma P}\sum_{i=1}^{\gamma}\sum_{p=1}^{P}(t_{ip} - c_i)^2. \tag{11}$$

The computation of this particular measure is linear in P and is thus more efficient to compute on-line during optimization code execution than other measures which are typically quadratic in P. In this application, the traits considered by the genetic algorithm are the (x, y) position of each sensor. As a practical consideration, we consider the genetic algorithm to be losing significant diversity once the diversity measure I reaches a value that is less than 50% of the initial value.

For the procedure of removing sensors, we follow a simple greedy heuristic algorithm. Specifically, at a given step i with m_i sensors $\{S_1, \ldots, S_{m_i}\}$ placed, we begin with proposing the removal of each sensor individually. Specifically, we develop m_i new proposed sets of $m_i - 1$ sensors each, where the sets are given as follows:

$$\{S_2, S_3, S_4, \ldots, S_{m_i}\}$$
$$\{S_1, S_3, S_4, \ldots, S_{m_i}\}$$
$$\vdots$$
$$\{S_1, S_2, \ldots, S_{m_i-2}, S_{m_i}\}$$
$$\{S_1, S_2, \ldots, S_{m_i-2}, S_{m_i-1}\}$$

and for each proposed new set we compute the resulting coverage C_W. The set of sensors with the highest coverage C_W is chosen as it provides the least impact on the system for removing a sensor. We thus remove that sensor from the list, re-number of the order, and repeat the procedure. This process stops when the highest coverage C_W that can be achieved for removal no longer meets the coverage constraint of $C_W \geq C_{min}$. When that happens, we do not remove any more sensors and the sensor removal heuristic completes. This process is a greedy approach to incrementally removing the most redundant sensors and will remove as many sensors as allowed to maintain the coverage constraint. Since no sensor placements are changed in this procedure, the service length L can only be decreased under the removal heuristic.

V. NUMERICAL RESULTS

To illustrate the performance of the proposed optimization procedure, we execute the optimization strategies on two simulated sensor placement example problems. The first problem we consider consists of a square domain $\mathcal{W}$ of size 100 units by 100 units. There are a maximum of $m_{max} = 36$ sensors to place in the domain to maintain a Q-coverage quality of at least $C_{min} = 98.0\%$. Each sensor has a coverage radius of $r_d = 22$ units (such that the coverage region $\mathcal{W}_i$ of an individual sensor is given by a disk of radius 22 units) with a coverage quality of $p_d = 0.95$ within $\mathcal{W}_i$. The coverage for this example is independent of the sensor position in the space, modeling a homogeneous environment where sensor placement has no effect on r_d nor p_d. Since each sensor covers to only 95% quality and the problem requires $C_{min} = 98\%$ overall quality, overlap must occur between the sensors, making the problem significantly more complicated than simple packing problems. This sensing problem employs a probabilistic sensing model with fixed probability over a fixed range, extensions to other probabilistic sensing models (such as [47]) are a straightforward numerical extension. These examples are performed to illustrate the tour length optimization problem under a Q-coverage constraint, extensions to other objectives and constraints are a subject of future work.

For the genetic algorithm, we apply standard binary encoding of the sensor positions in two-dimensional space. We represent each sensor's location $x_i \in \mathbb{R}^2$ by a 10-bit binary string, the first 5 bits of which correspond to its position on the x-axis and the second 5 bits correspond to its position on the y-axis. Thus, there are $2^5 = 32$ potential positions along each axis, corresponding to a resolution of 3.125 spatial units for positioning. The genetic algorithm is run with a population size of $P = 50$ individuals, and we apply a standard single-point crossover operator with a mutation rate of 2%. In addition, an elitist strategy of keeping the best population member from each generation is adopted, and remaining members are created by crossover (98% crossover rate). The inertia I as shown in equation (11) is computed at the end of every generation, yet we only consider the genetic algorithm stopping criteria at every tenth generation. This was chosen to avoid premature stopping in cases where the inertia drops significantly at a single generation (due to random effects in the population after mating) as opposed to following the general trend in inertia over multiple generations.

The $m = 36$ initial sensor placements are configured according to a regular grid at locations given by $x = [x_{val}, y_{val}]^T$ where

$$x_{val} \in \{\Delta/2, 3\Delta/2, 4\Delta/2, \ldots, (\sqrt{m}-1)\Delta/2\}$$

and

$$y_{val} \in \{\Delta/2, 3\Delta/2, 4\Delta/2, \ldots, (\sqrt{m}-1)\Delta/2\}.$$

In this expression, $\Delta = L/\sqrt{m}$ is the initial sensor-to-sensor spacing for a domain $\mathcal{W}$ of size $L \times L$ (Note: for this example, we have $\Delta = 100/\sqrt{36} = 16.67$). To generate some diversity in the initial population, we keep one copy of the initial configuration as given, and make $P-1$ perturbed copies, where the perturbation is independently applied to each x_{val} and y_{val} value, according to $x_{val} \rightarrow x_{val}(1+\varepsilon)$ for $\varepsilon \in U(-0.02, 0.02)$, and similarly for y_{val}. For this size of problem, that level of perturbation over $m = 36$ initial sensors provides for a level of diversity given by an inertia value of $I \approx 4.0$ (where I is computed as given in equation (11)). The initial configuration along with its corresponding optimal service route are shown in Fig. 3.

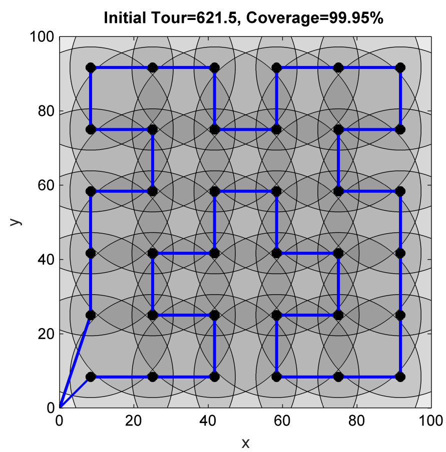

FIGURE 3. Initial configuration for the homogeneous environment examples. Sensor locations are represented as dots, the thick line represents the optimal service route of length L and the shaded circles represent the sensor coverage regions.

The description of one particular realization of applying the move-first optimization strategy (as outlined in the

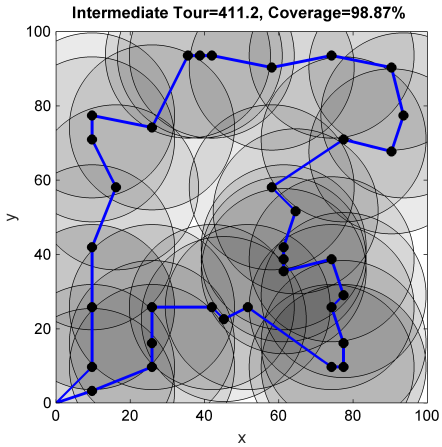

FIGURE 4. Configuration obtained from moving the $m = 36$ sensor positions in Fig. 3 to reduce the service route length L.

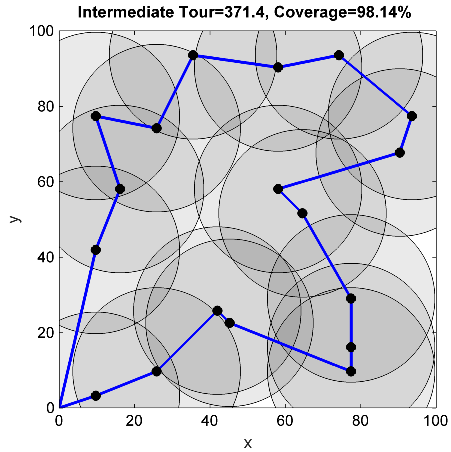

FIGURE 5. Configuration obtained from heuristic removal of 18 redundant sensors from the configuration shown in Fig. 4.

flowchart in Fig. 1) follows. When applied to this initial configuration, that algorithm allowed repositioning of the $m = 36$ sensors for 50 generations until the stopping criteria of a 50% reduction in inertia I was reported. The configuration after 50 generations is shown in Fig. 4, where it can be clearly seen that the service route length L is reduced by moving the sensors to a rough circle that is pushed out near the corners (to maintain coverage). At this point only the best configuration from the genetic algorithm is kept and the greedy heuristic for redundant sensor removal is performed. The greedy heursitic removed 18 of the 36 sensors, leaving a pattern of $m = 18$ sensors as shown in Fig. 5. From this configuration, we restart the genetic algorithm by again creating a new population of $P = 50$ individual configurations, where 49 of the configurations are simple perturbations of the configuration shown in the figure. Running the genetic algorithm to reposition these $m = 18$ sensors leads to a configuration as shown in Fig. 6 after an additional 50 generations. In this case we do not achieve the reduction of 50% for the inertia I, but rather we terminate due to 100 total generations of the genetic algorithm. We again check for any redundancy with the sensor removal heuristic and find that all $m = 18$ sensors are required to meet the coverage requirement of $C_W \geq 98\%$. Thus we conclude that the configuration in Fig. 6 is the optimized configuration for this scenario.

In Figs. 7 and 8 we show the reduction in the diversity (measured by inertia I) and the convergence in the service route length L, respectively, for this example. As the optimization approach is stochastic, we ran multiple realizations on the same problem to generate performance statistics. Running 100 independent realizations of the move-first algorithmic approach on this homogeneous problem resulted in solutions summarized by the first line of Table 1. In the table, we report nonparametric statistics on the results (specifically, we show the quartiles of the service route length values L and the final sensor count m), as the 100 solutions did not fit a

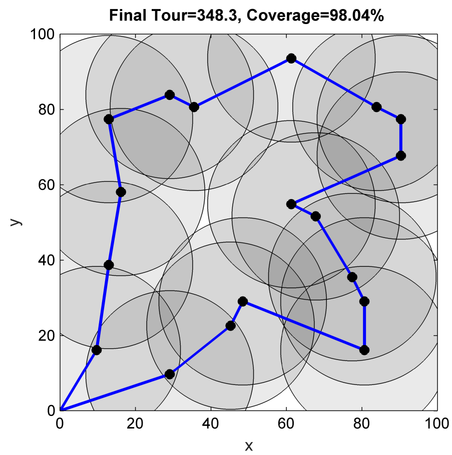

FIGURE 6. Configuration obtained from moving the $m = 18$ sensor positions in Fig. 5 to reduce the service route length L.

TABLE 1. Sensor configuration measures of the number of required sensors m and tour length L for various optimization approaches in the homogeneous environment.

Scenario	number of sensors m			tour length L			ρ
	Q_1	Q_2	Q_3	Q_1	Q_2	Q_3	
• move-first	17	18	19	327.2	335.8	343.5	0.34
• subsample-first	16	16	16	348.9	359.2	366.1	0.52
• heuristic (64)	n/a	14	n/a	n/a	378.8	n/a	n/a
• heuristic (100)	n/a	14	n/a	n/a	372.7	n/a	n/a

Note: For the stochastic methods, quartiles $\{Q_1, Q_2, Q_3\}$ over multiple runs are reported such that Q_2 refers to the medians.

measure of normality to statistical significance. We also compute the correlation ρ between the two statistical quantities m and L, and show that in the last column of Table 1. As the value of ρ is not large, we conclude that the statistics of m and L are independent, and that is why the quartiles $\{Q_1, Q_2, Q_3\}$ for each are reported separately.

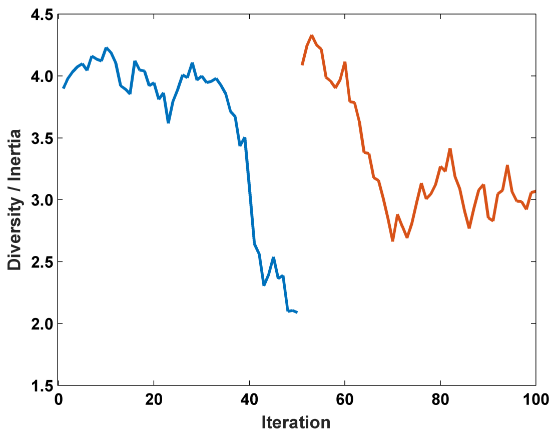

FIGURE 7. Diversity of the genetic algorithm population (as measured by the inertia I from equation (11)) for the move-first strategy in the example shown in Figs. 3 through 6. The line colors separate the different runs of the genetic algorithm that occur between sub-sampling stages.

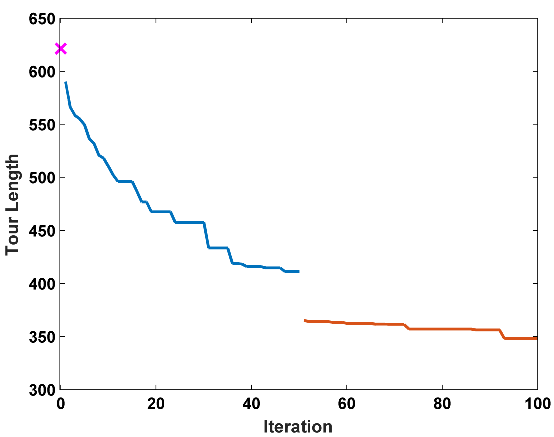

FIGURE 8. Convergence of the service route length objective L for the move-first strategy in the example shown in Figs. 3 through 6. The line colors separate the different runs of the genetic algorithm that occur between sub-sampling stages.

To demonstrate how the subsample-first strategy compares with the move-first strategy, we apply the subsample-first strategy to the same homogeneous environment example presented above. We again begin with $m = 36$ sensor locations on a regular grid for a homogeneous environment as shown in Fig. 3. For this strategy, we follow the flowchart in Fig. 2 that begins with the heuristic removal of redundant sensor locations. Results of the subsample-first strategy being run over 100 independent realizations is shown as the second row of Table 1. Comparing the results show comparable results with the move-first strategy, although the subsample-first does result in slightly fewer numbers of sensors m at the cost of a slightly larger service route length L. To compare these strategies with a more standard heuristic approach, we consider the greedy removal of sensors from an overpopulated grid of sensor locations to achieve the required coverage. The removal is continued until the coverage requirement is no longer met. It is different than our approach in that there is no changing of sensor positions, only removal from a more dense initial grid. We refer to this approach as ''heuristic'' in Table 1, and show results for initial grid sizes of $m_0 = 64$ and $m_0 = 100$. The heuristic approach yields smaller numbers of sensors than our new algorithm ($m = 14$ versus $m = 16$ or $m = 18$), yet the service tour length is larger by between 4% and 12%. Also, there is no improvement for further increasing the initial dense grid resolution as both $m_0 = 64$ and $m_0 = 100$ yield similar results. Thus, we conclude that our algorithm provides improved service tour lengths over the heuristic approach.

While the previous example illustrated the features of the optimization approach, it did not solve a physically realistic example. In practice, environmental conditions vary across the domain, and that may lead to areas of restriction (where sensors cannot be placed) or limited performance (where the sensor detection range is decreased). Since restricted locations can be readily modeled by setting the local r_d to zero, we consider both cases to be situations where the environment causes the sensor detection range to be dependent on the location x, such that $r_d \rightarrow r_d(x)$. In this way, changing a sensor's placement not only changes the location where its coverage $\mathcal{W}_i$ is applied, but it also changes the size of $\mathcal{W}_i$.

To demonstrate how the optimization approach works on such problems, we consider the placement of sensors in a heterogeneous environment that consists of a square domain $\mathcal{W}$ of size 100 units by 100 units, where the range $r_d(x)$ of the sensors is dependent on their location in the environment, according to the spatial range map shown in Fig. 9. The coverage radius of each sensor as shown in Fig. 9 is defined as a range for which the coverage quality is $p_d = 0.95$ within $\mathcal{W}_i$. There are a maximum of $m_{max} = 36$ sensors to place in the domain that is now designed to maintain a Q-coverage quality of at least $C_{min} = 95.0\%$. The initial position of the $m = 36$ sensors and their coverages are shown in Fig. 10. Note that the coverage in the upper left region is significantly smaller than the other parts of the domain, thus we expect that optimal configurations will have more dense sensor positioning in that subset of the domain. Also note that the service route is only dependent on the locations of the sensors, and thus is the same as shown in the previous example (see Fig. 3).

The description of one particular realization of applying the move-first strategy of Fig. 1 to the heterogeneous environment example follows. In this realization, the initial genetic algorithm led to repositioning of the $m = 36$ sensors for 20 generations until the genetic algorithm was stopped for diversity reduction. Then the redundant sensor removal heuristic removed 2 sensors and the remaining $m = 34$ sensors were repositioned using the genetic algorithm for another 50 generations (until the diversity reduction stopping criteria was met). From there another 2 sensors were removed using the redundant sensor removal heuristic and an additional 10 generations of the genetic algorithm were run to reposition the $m = 32$ sensors. From that point there was

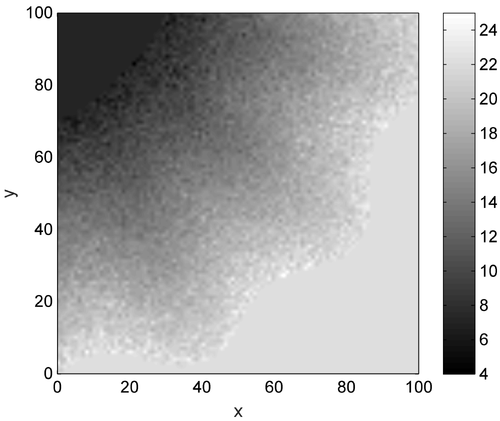

FIGURE 9. **Detection range map for the heterogeneous environment example. The range at each (x, y) position is represented by the color level.**

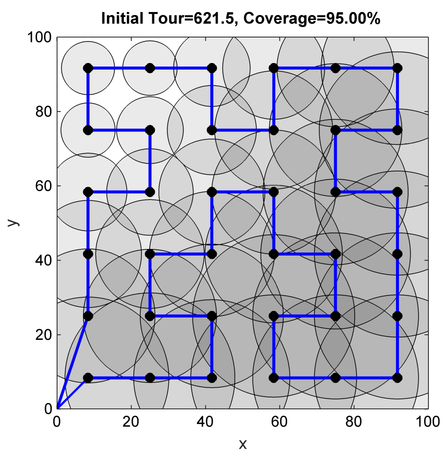

FIGURE 10. **Initial configuration for the heterogeneous environment example for the environment shown in Fig. 9.**

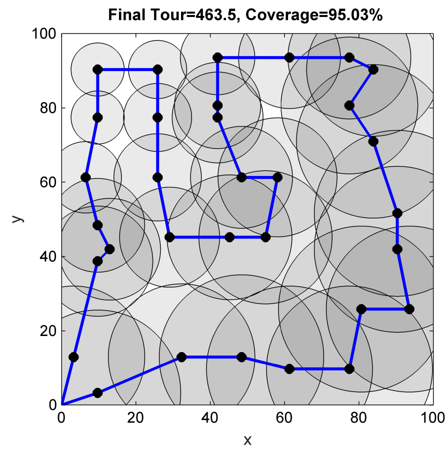

FIGURE 11. **Final configuration obtained from reducing the service length L using the move-first strategy for the heterogeneous environment example.**

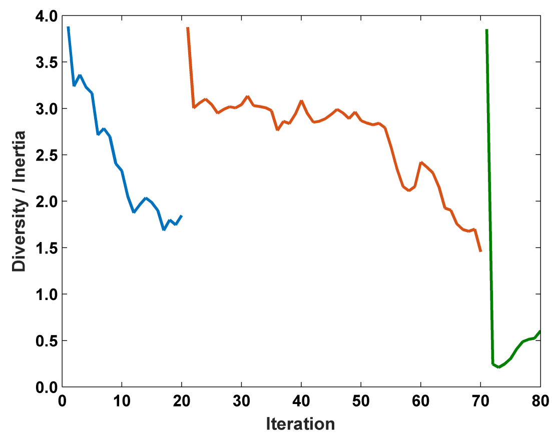

FIGURE 12. **Diversity of the genetic algorithm population (as measured by the inertia I from equation (11)) for the move-first strategy in the example shown in Figs. 10 and 11. The line colors separate the different runs of the genetic algorithm that occur between sub-sampling stages.**

no redundancy to be removed and the algorithm completed. The resulting configuration is shown in Fig. 11, where the top left of the configuration accepts much more back-and-forth motion in the service route in order to maintain coverage in the region of the domain with smaller r_d. The diversity measure and service length L convergence are shown in Figs. 12 and 13, respectively. Running 100 independent realizations of the move-first algorithmic approach on this heterogeneous problem results in solutions summarized by the first line of Table 2. As in the homogeneous case, the correlation ρ between the two statistical quantities m and L is low, so we report quartiles $\{Q_1, Q_2, Q_3\}$ independently to summarize the statistics.

We also apply the subsample-first strategy of Fig. 2 to the heterogeneous environment example and the resulting statistics from 100 independent realizations are presented in the second row of Table 2. In addition, the approach of heuristic removal of sensors from a densely populated grid of sizes $m_0 = 64$ and $m_0 = 100$ were also run for the heterogeneous case, and those results are shown in the final two rows of Table 2. From these results, it is seen that the two variants of our algorithm (move-first and subsample-first) both provide similar results with comparable mean values of m and L. However, the heuristic approach provided results with 3 fewer

TABLE 2. **Sensor configuration measures of the number of required sensors m and tour length L for various optimization approaches in the heterogeneous environment.**

Scenario	number of sensors m			tour length L			ρ
	Q_1	Q_2	Q_3	Q_1	Q_2	Q_3	
• move-first	27	28	30	453.7	463.3	472.3	0.13
• subsample-first	27	28	29	450.0	460.0	470.5	0.57
• heuristic (64)	n/a	25	n/a	n/a	487.9	n/a	n/a
• heuristic (100)	n/a	25	n/a	n/a	490.8	n/a	n/a

Note: For the stochastic methods, quartiles $\{Q_1, Q_2, Q_3\}$ over multiple runs are reported such that Q_2 refers to the median.

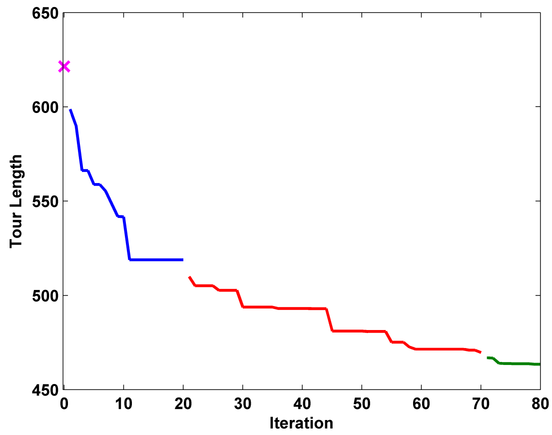

FIGURE 13. Convergence of the service route length objective L for the move-first strategy in the example shown in Figs. 10 and 11. The line colors separate the different runs of the genetic algorithm that occur between sub-sampling stages.

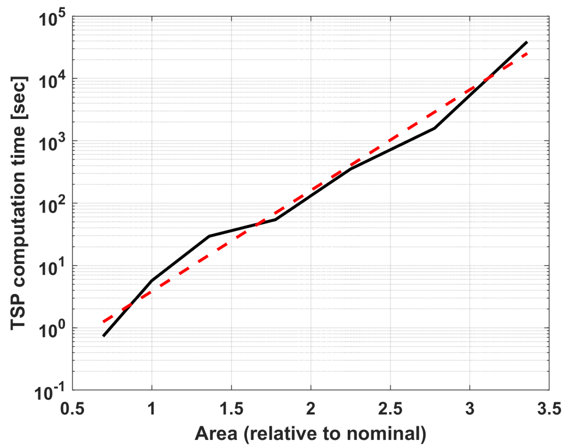

FIGURE 14. Computational run time for the initial serviceability (tour length) computation as a function of problem size. Solid black line is numerical results; red dashed line is an exponential curve fit. The nominal area size (given by an area size of one) corresponds to the examples in the paper.

sensors ($m = 25$ as compared to $m = 28$) but at the cost of an approximately 6% increase in service tour length. Thus, if service tour length is a priority, our algorithm once again provides improved results over the heuristic approach.

The examples that were presented above considered tens of sensors in a region. As the optimization approach embeds a traveling salesman problem within a genetic algorithm, the former of which is known to be NP-hard, there is a concern that the algorithm run-time may grow unreasonably large for increasing numbers of sensors (corresponding to larger applications). To address this issue, we have run a sequence of problems of increasing size, and computed how much additional computational time the TSP solution took as the problem size increased. The results of this study are shown in Fig. 14, where we plot the time spent on the TSP solution (which is the dominant portion of the method as the problem gets large) as a function of the problem size, which is given by the area of the region to be covered by the sensors (sensor range is fixed for this plot). It is clear from the figure that there is an exponential growth to the computation time, in fact, the dashed red curve shows an exponential fit to the TSP computation times t_{TSP} given by $t_{TSP} = 0.093 \exp(3.72\ A/A_0)$, where A_0 is the nominal area corresponding to the other examples given in this paper. Thus, for problems of the scale given in this paper, the computation can be computed within seconds. However, as the region gets to be even three times as large, the run-time grows to nearly 10,000 seconds (or 2.78 hours). Clearly for anything much larger than that, the run times may become unreasonable for some particular applications. However, as a planning tool, it may be reasonable to wait a significant amount of time to find the best configuration for placing sensors that are to be used for long-term monitoring applications.

The numerical approach that has been presented can be applied to other extensions of sensor networks for which optimizing the serviceability is desired. In particular, while the examples presented were for domains that were rectangular, the use of a location-dependent detection range $r_d(x)$ (as shown in the second example) allows the direct consideration of sub-regions of the space $\mathcal{W}$ to have $r_d(x) = 0$. In that way, any non-rectangular domain is handled by considering a bounding rectangle and "zeroing out" those portions of the bounding rectangle that are not in the domain. Additionally, features such as range-dependent sensors or directional sensors can be handled numerically by creating more complex regions $\mathcal{W}_i$ in the formulation in equation (2). The joint optimization of sensor look-angles in a directional sensor network would also necessitate the extension of the binary encoding of each sensor in the genetic algorithm to incorporate this extra variable. These topics are subjects of future work.

VI. CONCLUSION

A new method for optimizing the service route length for a sensor network, while maintaining a prescribed coverage demand, has been developed. The method uses an alternating optimization procedure, which alternates between optimizing the positions of a fixed number of sensors to reduce the service route length and removing any sensors that are redundant for the coverage demand. The sensor position optimization utilizes a genetic algorithm with a stopping criteria based on the diversity of the population, while the sensor removal procedure employs a greedy heuristic to remove redundant sensors that are not required to meet the coverage demand. This approach is shown to converge to similar solutions regardless of whether it starts with sensor removal or sensor repositioning. Numerical examples for both a homogeneous environment as well as a practical environment (i.e. an environment with sensor performance that has variability depending on sensor positions) were shown to illustrate the effectiveness of the approach, and it compares favorably against a heuristic approach.

REFERENCES

[1] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *Proc. 3rd Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2005, pp. 154–165.

[2] J. Polo, G. Hornero, C. Duijneveld, A. García, and O. Casas, "Design of a low-cost wireless sensor network with UAV mobile node for agricultural applications," *Comput. Electron. Agricult.*, vol. 119, pp. 19–32, Nov. 2015.

[3] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Deployment and connectivity repair of a sensor net with a flying robot," in *Experimental Robotics IX* (Springer Tracts in Advanced Robotics), M. H. Ang and O. Khatib, Eds. Berlin, Germany: Springer, 2006, pp. 333–343.

[4] R. Z. Farahani, N. Asgari, N. Heidari, M. Hosseininia, and M. Goh, "Covering problems in facility location: A review," *Comput. Ind. Eng.*, vol. 62, no. 1, pp. 368–407, 2012.

[5] Y. Gu, Y. Ji, J. Li, and B. Zhao, "QoS-aware target coverage in wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 9, no. 12, pp. 1645–1659, Dec. 2009.

[6] O. Berman, Z. Drezner, and D. Krass, "Cooperative cover location problems: The planar case," *IIE Trans.*, vol. 42, no. 3, pp. 232–246, Dec. 2009.

[7] O. Berman, Z. Drezner, and D. Krass, "Discrete cooperative covering problems," *J. Oper. Res. Soc.*, vol. 62, no. 11, pp. 2002–2012, Nov. 2011.

[8] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Comput. Surv.*, vol. 43, no. 4, p. 32, Oct. 2011.

[9] R. Ghazalian, A. Aghagolzadeh, and S. M. H. Andargoli, "Energy optimization of wireless visual sensor networks with the consideration of the desired target coverage," *IEEE Trans. Mobile Comput.*, early access, Apr. 27, 2020, doi: 10.1109/TMC.2020.2990596.

[10] S. Halder and A. Ghosal, "A location-wise predetermined deployment for optimizing lifetime in visual sensor networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 6, pp. 1131–1145, Jun. 2016.

[11] H. Liu, X. Chu, Y.-W. Leung, and R. Du, "Minimum-cost sensor placement for required lifetime in wireless sensor-target surveillance networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1783–1796, Sep. 2013.

[12] D. S. Deif and Y. Gadallah, "Classification of wireless sensor networks deployment techniques," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 834–855, 2nd Quart., 2014.

[13] F. Y. S. Lin and P. L. Chiu, "A near-optimal sensor placement algorithm to achieve complete coverage/discrimination in sensor networks," *IEEE Commun. Lett.*, vol. 9, no. 1, pp. 43–45, Jan. 2005.

[14] X. Liu, "A deployment strategy for multiple types of requirements in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2364–2376, Oct. 2015.

[15] Q. Wu, N. S. V. Rao, X. Du, S. S. Iyengar, and V. K. Vaishnavi, "On efficient deployment of sensors on planar grid," *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2721–2734, Oct. 2007.

[16] J. H. Jaramillo, J. Bhadury, and R. Batta, "On the use of genetic algorithms to solve location problems," *Comput. Oper. Res.*, vol. 29, no. 6, pp. 761–779, May 2002.

[17] Y. Yoon and Y.-H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1473–1483, Oct. 2013.

[18] A. Singh, A. Rossi, and M. Sevaux, "Matheuristic approaches for Q-coverage problem versions in wireless sensor networks," *Eng. Optim.*, vol. 45, no. 5, pp. 609–626, May 2013.

[19] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1448–1453, Dec. 2002.

[20] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos, "Efficient sensor placement optimization for securing large water distribution networks," *J. Water Resour. Planning Manage.*, vol. 134, no. 6, pp. 516–526, Nov. 2008.

[21] E. Piña-Covarrubias, A. P. Hill, P. Prince, J. L. Snaddon, A. Rogers, and C. P. Doncaster, "Optimization of sensor deployment for acoustic detection and localization in terrestrial environments," *Remote Sens. Ecology Conservation*, vol. 5, no. 2, pp. 180–192, Jun. 2019.

[22] J. Jia, J. Chen, G. Chang, and Z. Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm," *Comput. Math. with Appl.*, vol. 57, nos. 11–12, pp. 1756–1766, Jun. 2009.

[23] J. Yu, Y. Chen, L. Ma, B. Huang, and X. Chen, "On connected target k-coverage in heterogeneous wireless sensor networks," *Sensors*, vol. 16, no. 1, p. 104, 2016.

[24] S.-K. Gupta, P. Kuila, and P. K. Jana, "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks," *Comput. Elect. Eng.*, vol. 56, pp. 544–556, Nov. 2016.

[25] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 550–586, 1st Quart., 2017.

[26] G. Molina, F. Luna, A. J. Nebro, and E. Alba, "An efficient local improvement operator for the multi-objective wireless sensor network deployment problem," *Eng. Optim.*, vol. 43, no. 10, pp. 1115–1139, 2011.

[27] S. Sengupta, S. Das, M. D. Nasir, and B. K. Panigrahi, "Multi-objective node deployment in WSNs: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity," *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 405–416, 2013.

[28] T. A. Wettergren and R. Costa, "Optimal multiobjective placement of distributed sensors against moving targets," *ACM Trans. Sensor Netw.*, vol. 8, no. 3, pp. 21:1–21:23, 2012.

[29] M. Ma, Y. Yang, and M. Zhao, "Tour planning for mobile data-gathering mechanisms in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1472–1483, May 2013.

[30] D. G. Macharet, J. W. G. Monteiro, G. R. Mateus, and M. F. M. Campos, "Bi-objective data gathering path planning for vehicles with bounded curvature," *Comput. Oper. Res.*, vol. 84, pp. 195–204, Aug. 2017.

[31] M. Ma and Y. Yang, "SenCar: An energy-efficient data gathering mechanism for large-scale multihop sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 10, pp. 1476–1488, Oct. 2007.

[32] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," in *Proc. IEEE Int. Symp. Parallel Distrib. Process.*, Apr. 2008, pp. 1–9.

[33] I. Khoufi, E. Livolant, P. Minet, M. Hadded, and A. Laouiti, "Optimized trajectory of a robot deploying wireless sensor nodes," in *Proc. IFIP Wireless Days (WD)*, Nov. 2014, pp. 1–6.

[34] W. L. Chang, D. Zeng, R. C. Chen, and S. Guo, "An artificial bee colony algorithm for data collection path planning in sparse wireless sensor networks," *Int. J. Mach. Learn.*, vol. 6, no. 3, pp. 375–383, 2015.

[35] G. Yogarajan and T. Revathi, "Nature inspired discrete firefly algorithm for optimal mobile data gathering in wireless sensor networks," *Wireless Netw.*, vol. 24, no. 8, pp. 2993–3007, Nov. 2018.

[36] A. Kaswan, A. Tomar, and P. K. Jana, "An efficient scheduling scheme for mobile charger in on-demand wireless rechargeable sensor networks," *J. Netw. Comput. Appl.*, vol. 114, pp. 123–134, Jul. 2018.

[37] Z. Lyu, Z.Wei, J. Pan, H. Chen, C. Xia, J. Han, and L. Shi, "Periodic charging planning for a mobile WCE in wireless rechargeable sensor networks based on hybrid PSO and GA algorithm," *Appl. Soft Comput.*, vol. 75, no. 1, pp. 388–403, Feb. 2019.

[38] S. H. Kang, S. J. Kim, and H. Zhou, "Optimal sensor positioning; A probability perspective study," *SIAM J. Scientific Comput.*, vol. 39, no. 5, pp. B759–B777, Jan. 2017.

[39] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton, NJ, USA: Princeton Univ. Press, 2007.

[40] G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson, "Solution of a large-scale traveling salesman problem," *J. Oper. Res. Soc. Amer.*, vol. 2, no. 4, pp. 393–410, 1954.

[41] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *J. ACM*, vol. 7, no. 4, pp. 326–329, 1960.

[42] U. Pferschy and R. Staněk, "Generating subtour elimination constraints for the TSP from pure integer solutions," *Central Eur. J. Oper. Res.*, vol. 25, no. 1, pp. 231–260, 2016.

[43] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley, 1989.

[44] C. Mattiussi, M. Waibel, and D. Floreano, "Measures of diversity for populations and distances between individuals with highly reorganizable genomes," *Evol. Comput.*, vol. 12, no. 4, pp. 495–515, 2004.

[45] R. W. Morrison and K. A. De Jong, "Measurement of population diversity," in *Artificial Evolution* (Lecture Notes in Computer Science),vol. 2310, P. Collet, C. Fonlupt, J.-K. Hao, E. Lutton, and M. Schoenauer, Eds. Berlin, Germany: Springer, 2002, pp. 31–41.

[46] T. A. Wettergren and R. Costa, "Efficient genetic algorithms for covering demand points," *Int. J. Metaheuristics*, vol. 4, nos. 3–4, pp. 244–267, 2015.

[47] H.-L. Wang and W.-H. Chung, "The generalized k-coverage under probabilistic sensing model in sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 1737–1742.

THOMAS A. WETTERGREN (Senior Member, IEEE) received the B.S. degree in electrical engineering and the Ph.D. degree in applied mathematics from the Rensselaer Polytechnic Institute, Troy, NY, USA. He is currently working as a Research Scientist with the Naval Undersea Warfare Center, Newport, RI, USA, where he serves as the U.S. Navy Senior Technologist (ST) for Operational and Information Science. He is an Adjunct Professor of industrial and systems engineering with the University of Rhode Island. He is the author of the book *Information-Driven Planning and Control* (MIT Press, 2021). His research interests include the development of new analytical and computational methods for the mathematical modeling, optimal planning, and adaptive control of distributed groups, with particular focus on groups of sensors and unmanned vehicles. He was a recipient of the NAVSEA Scientist of the Year, the Assistant Secretary of the Navy Top Scientists of the Year, and the IEEE-USA Harry Diamond Award. He is a member of SIAM.

RUSSELL COSTA (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Massachusetts, Dartmouth, North Dartmouth, MA, USA, and the Ph.D. degree in electrical engineering from the University of Rhode Island, Kingston, RI, USA. He is currently a Research Engineer with the Naval Undersea Warfare Center, Newport, RI, USA, where he serves as the Chief Scientist of the Ranges, Engineering, and Analysis Department. He is a member of a Multi-Organizational Mathematical Optimization Team, where he is collaborating with other researchers on development of novel mathematical techniques in undersea distributive sensing. He has published six refereed journal articles and eight conference papers. His current research interests include applied optimization, target motion analysis, information geometry, probabilistic modeling, and distributed simulation. He has been a Reviewer for *ACM Transactions on Sensor Networks* and IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS journals.