Communication Coverage for Independently Moving Robots

Stephanie Gil, Dan Feldman, Daniela Rus

Abstract—We consider the task of providing communication coverage to a group of sensing robots (sensors) moving independently to collect data. We provide communication via controlled placement of router vehicles that relay messages from any sensor to any other sensor in the system under the assumptions of 1) no cooperation from the sensors, and 2) only sensor-router or router-router communication over a maximum distance of R is reliable. We provide a formal framework and design provable exact and approximate (faster) algorithms for finding optimal router vehicle locations that are updated according to sensor movement. Using vehicle limitations, such as bounded control effort and maximum velocities of the sensors, our algorithm approximates areas that each router can reach while preserving connectivity and returns an expiration time window over which these positions are guaranteed to maintain communication of the entire system. The expiration time is compared against computation time required to update positions as a decision variable for choosing either the exact or approximate solution for maintaining connectivity with the sensors on-line.

I. Introduction

We wish to provide communication coverage to mobile sensing robots, or "sensors", moving autonomously according to unknown trajectories by using a team of routing vehicles whose movement we can control. We are interested in maintaining a network such that each sensor can send and receive messages to and from all other sensors in the system but we do not assume sensor cooperation with the routers to maintain the network. This is an important assumption as it a) allows the mobile sensors maximum freedom to change their motion plans as necessary, for example when exploring unknown environments for search and rescue missions, and b) does not require sensors to maintain any information about other sensors in the network, similar to cell phone clients who can reach anyone in their network without having to know their locations. However, this also makes communication maintenance significantly more challenging. In this spirit, we assume that 1) every router can communicate with any other router or sensor at a distance of at most R, where R is a specified communication radius and 2) a sensor can only communicate reliably with its nearest router.
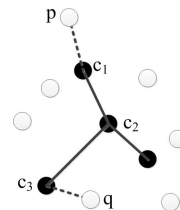
Fig. 1: The sensor p can communicate with a sensor q through the path of router vehicles (centers) p, c_1, c_2, c_3, q. The minimum required transmission power r for this communication is the longest edge (c_2, c_3) in this path.

We aim to develop on-line position control for the router vehicles such that they provide communication over the entire system, and such that the amount of time that these positions are guaranteed to maintain a connected network is maximized for the given communication radius and vehicle velocities.

In order to achieve the goal of controlling router vehicles to provide communication under these assumptions we must address questions such as, given n mobile sensors can we allocate k mobile routers such that there exists a connected communication network? How would we compute this allocation and moreover how can we maximize the amount of time that we are guaranteed to preserve mutual connectivity despite unpredicted sensor movement? What are the tradeoffs between the communication radius R, number of mobile routers k, maximum sensor velocities, and their moving freedom? In this work we define a formal framework, design provable algorithms, and provide empirical case studies that aim to answer these types of questions.

A. Summary of Problems Addressed

For a communication radius of R, a set C of router vehicles, and a set S of sensors, communication is *feasible* if every pair of sensors can mutually communicate using C. Communication occurs via message passing from a sensor p to its nearest router c_1, and then via multi-hop through the connected network of routers $c_2, c_3, \cdots$ before being delivered to its destination sensor q. The communication radius is the longest communication edge between any two vehicles in this routing path; see Fig. 1. We consider communication to be feasible for a configuration C if every pair of sensors p, q have such a path that requires a communication radius of no more that R. This constitutes our first problem.

Problem 1 (feasibility problem): Given $R > 0$, a

configuration C and S of router and sensor positions, is communication *feasible*?

Note that the solution depends on all the possible spanning trees of the set C. If the required communication radius is actually r where $r < R$, this implies that sensors p and q can move while still preserving mutual connectivity which motivates the problem of finding the *minimum* such r:

Problem 2 (k-connected center problem): What is the minimum value of R such that communication is feasible for a set of router positions C (centers) given a set of sensor positions S, and what is that C?

We note that even if the number of centers and mobile sensors is the same $(n = k)$ the solution is not trivial due to the interdependencies arising from the connectivity requirement of the centers. Finally, we incorporate the dynamics of the vehicles, i.e., control effort limitations for the routers and maximum velocities for the mobile sensors, as well as the maximum communication radius R to answer the following question.

Problem 3 (reachability problem): Suppose that we are given the current position of a set C of routers, their allowed control inputs, current sensor positions S and their maximum sensor velocities. What areas can each sensor and router move to (reach) while keeping the network connected? For how long can we guarantee that a connected configuration is maintainable?

B. Results Snapshot

We develop algorithms that compute router vehicle positions that provide a connected communication network if such a configuration exits, and that maximize a sufficient condition for reachability for given vehicle velocities. The feasibility problem can be answered in $O(n + (k \log k)^{4/3})$ for sensors in $\mathbb{R}^3$ and $O(n + (k \log k))$ in $\mathbb{R}^2$ for a given C by the observation that an Euclidean minimum spanning tree for C (that minimizes the sum of the length of edges) also minimizes the longest edges among all spanning trees of C [1]. For n mobile sensors and k centers our algorithms provide: 1) an exact solution C^* with minimum communication cost h^* in $n^{O(k)}$ time, or 2) a faster approximate solution that takes $(\frac{k}{\epsilon})^{O(k)}$ time and returns a solution C with cost h such that $h \leq (1 + \epsilon)h^*$, where $\epsilon > 0$ is an input parameter depending on the desired solution accuracy. We provide an *expiration time*, or time window, over which a configuration C of communication vehicle positions is guaranteed to maintain connectivity of the network given control input limitations and maximum sensor velocities. This expiration time compared against required computation time provides a lower bound estimate of whether there is enough time to compute the exact solution, or whether the approximation algorithm should be used (see Fig. 3). Intuitively, for large systems or for cases where the positions of the routers must be updated

quickly, the faster approximation algorithm is used. More generally, we show that α-approximations for the traditional k-centers problem are $(3 + 2\alpha)$ approximations for the k-connected centers problem. Since there exist fast $O(nk)$ time 2-approximation algorithms for the k-center problem, we thus obtain 7-approximation algorithm for the connected k-center problem as well.

Our empirical results show that for a scenario of 5 sensors moving at a constant $1m/\text{sec}$ and 2 routing vehicles moving at $1.5m/\text{sec}$, the expiration time is $t_e > 60\text{sec}$ and the minimum communication cost is computed in 38sec using Matlab and CVX [2] on an Intel Core 2 Duo 2.4GHz processor. As sensors move towards the communication limits a switch is made to the approximate version of the algorithm which computes a solution in 3sec with cost that is only 13% larger than the minimum communication cost.

C. Related Work

Both centralized and decentralized approaches for connectivity maintenance amongst single or multiple moving robots have been investigated in the literature, mostly where control or cooperation of vehicles in the system is assumed [3]–[6]. Connectivity for an adversarial agent is investigated in [7] for a single agent that must be tethered to a base station. The authors' previous work, [8] and [9], provides distributed approaches for communication and connectivity problems but in that work sensors are assumed to be stationary. The paper by [10] is more closely related to our current approach in that reachability between agents is investigated. However the element of uncontrolled motion in the current paper precludes fixed communication assignments between agents that can be maintained throughout. Instead, we leverage results from computational geometry to handle these changes in topology. This paper uses reductions inspired by the classic k-center problem, which, as noted in previous section, has a simple greedy 2-approximation algorithm in $O(nk)$ time [11]. Coresets can be viewed as a general approximation concept for more efficient computation of NP hard problems and includes applications from random sampling, feature extraction, and more. See a comprehensive survey on this topic by Agarwal, Har-Peled, and Varadarajan [12]. The coresets that most relevant in our context are for the k-center problem [13] and for bi-criteria approximations [14]. Reachability techniques have been used for control of complex dynamic systems [15], [16], and in achieving of feasibility of a target state over long or infinite horizons as in the seminal work of [17].

II. PROBLEM STATEMENT

We wish to provide communication coverage to n mobile sensors with positions at time t, $s_{j_t} \in \mathbb{R}^d$, that are moving over unknown trajectories. We assume the

following model for sensor position updates, the input w is unknown but bounded to the uncertainty set W:

$$s_{j\,t+1} = s_{j\,t} + w_j, \ j \in \{1, \ldots, n\}, \tag{1}$$

$$w_j \in W = \{w | \ \|w\| \le v_S\}. \tag{2}$$

We provide this communication using a set of mobile router vehicles with positions $c_{i\tau} \in \mathbb{R}^d$, $i \in \{1, \ldots, k\}$, whose movement we can control via the input u. The set U specifies the vehicle control limitations:

$$c_{i\,\tau+1} = c_{i\tau} + u_{i\tau}, \ i \in \{1, \ldots, k\}, \tag{3}$$

$$u_{i\tau} \in U = \{u | \ \|u\| \le v_C\} \tag{4}$$

We use different notations t and τ where t is time in seconds, and τ is the time in number of iterations for any algorithm that updates the router positions. We can think of iterations as a sequence of times $(\tau_1, \tau_2, \ldots)$ where $\tau_{i+1} = t_i + B$ and t_i is the time in seconds of the current position update for the router vehicles and B is the elapsed time in seconds until the next update. As a shorthand, we will use the notation τ and $\tau + 1$ to denote two consecutive updates. We denote the set of router positions at iteration τ as C_τ and the set of sensor positions at time t as P_t, and assume d is a constant. For brevity and clarity in the remainder of the paper we use the names *router vehicle* and *centers* synonymously.

We would like to keep the heterogeneous system in a *connected configuration*. In order to precisely define a connected configuration, we first introduce a *minimum bottleneck spanning tree* or MBST

Definition 2.1 (MBST): Given a set $C \subseteq \mathbb{R}^d$, a *spanning tree* of C is a tree $G(C, E)$ that connects all the centers (points) in C, whose edges are $E \subseteq C^2$. A *bottleneck edge* is the longest edge in a spanning tree, i.e, that maximizes $\text{dist}(c, c')$ over $(c, c') \in T$. A spanning tree T^* is a *minimum bottleneck spanning tree (or MBST)* of C, if C does not contain a spanning tree with a shorter bottleneck edge. We define $b(C)$ to be the length of the bottleneck edge of the MBST of C, i.e,

$$b(C) := \max_{(c,c') \in T^*} \text{dist}(c, c'). \tag{5}$$

The length of the minimum bottleneck edge $b(C)$, and the maximum distance between any sensor to its closest center determine the smallest communication radius, r^*, needed to achieve a connected system. If this value is smaller than the maximum allowed communication radius R, then the configuration is connected. Formally,

Definition 2.2 (Connected Configuration): Let $P \subseteq \mathbb{R}^d$ and $k \ge 1$ be an integer. Let

$$r(P, C) := \max_{s \in P} \text{Dist}(s, C) \tag{6}$$

where $\text{Dist}(s, C) = \min_{c \in C} \text{dist}(s, c)$ is the closest point to s in the set C. For a given set C, define

$$rb(P, C) = \max\{r(P, C), b(C)\} \tag{7}$$

If $rb(P, C) \le R$ then sets P and C are in a connected configuration. We denote the set of connected configurations for P and C as $\Omega(P, C)$. Moreover, if a connected configuration for a set P exists for some C

then a connected configuration is *feasible*.

We would like to address the problems of 1) evaluating whether a connected configuration is feasible for a set P and C, 2) finding the minimum value $rb(P, C)$ such that the system is connected and 3) choosing a new set positions $C_{\tau+1}$ such that the centers maintain feasibility of a connected configuration from one iteration C_τ to the next $C_{\tau+1}$ given the vehicle dynamics models from (4) and (2) and the communication radius R. We formalize the third question outlined above by using terminology from reachability theory. In particular, we define reachability of a set X^*

Definition 2.3 (Reachability of a set X^*): We use the definition of reachability of a set from [17] for a one-step horizon where a sufficient condition for X^* to be reachable from a state x (where the state evolution is given by equations of the form (2) and (4)) is $x \in X$ where X is defined as

$$\mathcal{E} = \{x | x + w \in X^* \forall w \ s.t. \ \|w\| \le v_S\} \tag{8}$$

$$X = \{x | x + u \in \mathcal{E} \text{ for some } u, \ \|u\| \le v_C\} \tag{9}$$

Specifically, the condition that $x \in X$ asserts that for all uncontrolled but bounded inputs w to the system where $\|w\| \le v_S$, there exists some permitted control $u, \|u\| \le v_C$, such that the state x can remain in the desired set X^*.

Therefore we wish to choose $C_{\tau+1}$ such that a connected configuration $\Omega(P_{\tau+1}, C_{\tau+1})$ is reachable. This formulation also allows us to identify and maximize the *expiration time* for a set of positions C:

Definition 2.4 (Expiration time t_e): For given problem parameters v_S, v_C, R, and current vehicle positions P, C, we define the expiration time

$$t_e = T_E(P, C, v_S, v_C, R) \tag{10}$$

to be a lower bound on the time window over which the set of positions C is guaranteed to maintain C and $\tilde{P}$ in a connected configuration, for any set of positions $\tilde{P}$ that evolves from P using the update equations in (2). Using the expiration time, we choose our update time according to $\tau_{l+1} - \tau_l \le t_e$.

We answer questions about feasibility and minimum radius r in Section III-A, and questions about reachability and expiration time in Section III-B.

III. Controller Development

In this section we derive an algorithm for finding optimal placement for the k centers that provides a connected configuration for n mobile sensors assuming only sensor to center and center to center communication. This algorithm is centralized and assumes access to all current sensor positions at each update time in order to allow maximum flexibility in network structure change due to sensor movement. We develop a novel formulation for this problem, namely the k-connected centers problem. We use tools from reachability analysis

to incorporate unknown sensor motion, control input limitations for the router vehicles, and communication range specifications to provide a solution that is guaranteed to be both *connected* and *feasible* over a computable time window, t_e, that we maximize.

A. k-Connected Centers

The k-center is a set C^* of k centers (points) in $\mathbb{R}^d$ that minimizes, among all such k centers, the maximum distance from every point P to its nearest center in C^*. Unfortunately, k-centers does not solve the problem of providing a fully connected network over all sensors and communication vehicles. We modify the k-center problem to also minimize the maximum inter-communication vehicle distance. Since it is prohibitive and unnecessary to constrain the communication vehicles to a fully connected system, we instead define a spanning tree over this set:

Definition 3.1 (k-Connected Centers): The k-connected centers is a set C^* that minimizes

$$rb(P,C) := \{\max\{r(P,C), b(C)\}\}$$

for a given input set P over every set C of k points in $\mathbb{R}^d$, where r is defined in (6), and $b(C)$ from Equation (5) is defined over all spanning trees.

Once connectivity assignments are made between sensors and centers (for example, the two outer loops in Algorithm 1), we can define a connectivity neighborhood:

Definition 3.2 (Connectivity Neighborhood): For a given configuration of centers C_τ at iteration τ, a center c_i has an assignment Q_i of sensors where $Q_i \subset P$ and possibly a connectivity edge with one other center c_j such that $c_j = T(c_i)$. The notation $T(c_i)$ is used to denote connectivity constraints between centers and is a *directed* constraints meaning that $T(c_i) = c_j$ does not imply $T(c_j) = c_i$. In fact we avoid this in order to prevent dependency cycles. The connectivity neighborhood $\mathcal{N}(c_i)$ of c_i is the set of vehicles that a communication vehicle is assigned to maintain connectivity to. Although connectivity constraints and neighborhoods are directed, communication is *undirected*, determined only by inter-vehicle distance.

B. Reachability

The solution to the k-connected sensors problem does not account for uncertain sensor movement which may render a connected configuration infeasible given the control limit placed on the centers. Our key insight is to abstract the *a priori* unknown sensor movement as a disturbance on the system of communication vehicles and apply tools from reachability analysis. Taking this perspective, we attempt to use control of the communication vehicles to maintain a connected configuration over a span of t_e seconds which is an *expiration time* window that we maximize. We note that reachability

tools are typically applied over a horizon of more than 1, however our focus is to provide sensors maximum freedom in their motion and thus we allow for changes in network topology at each iteration.

We derive the form of the reachability matrix K for our problem. For a general derivation of this reachability matrix the reader is referred to [17]. We define our target set as $X^* = \{x| \ \|x\| \leq R\}$ where x is the relative state between a center and its connectivity neighbor from Definition 3.2, specifically, $x = (p - c_i)$ for $p \in \mathcal{N}(c_i)$. Therefore $(p - c_i) \in X^*$ implies that c_i and its connectivity neighbor p are connected. The idea is the following: given a sensor position s_t and a disturbance set W from (2), we want to find all the locations for a center c such that the relative state $(s_t - c) + W \subset X^*$. In particular we want to know all c for which $(s_t - c) \in \mathcal{E}_{t+1}$ where

$$\mathcal{E}_{t+1} = \{x| \ x + w_t \in X^* \ \ \forall w_t \in W\} \qquad (11)$$

This set represents all the locations for a center such that regardless of the sensor trajectory, connectivity is maintained. Additionally, we must account for the control limitations on our communication vehicles. Not all positions c such that $(c - s_t) \in \mathcal{E}_{t+1}$ are attainable by a center with position c_t at time t due to the constraint that $\|c_t - c_{t+1}\| \leq v_C$. Therefore the reachability set X takes the form:

$$X_t = \{x_t| \ x_t + u_t \in \mathcal{E}_{t+1} \text{ for some } u_t \in U\} \quad (12)$$

Where U is the set of allowable control actions from Equation (4). While X_t is a convex set, it is not ellipsoidal in general. Since ellipsoidal sets provide simple quadratic constraints, an approximation to this set is desirable for computation. In [17] it is shown that the set X_t has an inner ellipsoidal approximation $\bar{X}_t \subset X_t$ and that $(s_t - c_{t+1}) \in \bar{X}_t$ is a sufficient condition for connectivity in the next timestep such that $(s_{t+1} - c_{t+1}) \in X^*$. Therefore if for every center c_{i_t} at time t, we satisfy $(p_t - c_{i_{t+1}}) \in \bar{X}_t$ for all connectivity neighbors $p_t \in \mathcal{N}(c_{i_t})$ then we maintain connectivity over the entire system of centers and sensors over the next iteration.

The set $\bar{X}_t$ can be described by the $d \times d$ *reachability matrix* K. Problems with velocity bounds of the form (4) and (2) permit a diagonal structure of the matrix K from which we can derive a simple formula for assessing the feasibility of maintaining a connected network for any C and P given the problem parameters v_S, v_C and R:

$$\bar{X}_t = \{x_t| \ x_t^T K x_t \leq 1\}, K = diag(\nu) \qquad (13)$$

$$\nu = \begin{cases} \frac{1}{(1-\beta)(R^2 - \frac{1}{\beta}v_S^2) + v_C^2} & \text{if } x_t = s_t - c_t \\ \frac{1}{(1-\beta)(R^2 - \frac{1}{\beta}v_C^2) + v_C^2} & \text{if } x_t = T(c)_t - c_t \end{cases}$$

Where $0 < \beta < 1$ is an approximation constant. We can now formally define the reachable k-connected centers problem:

Definition 3.3 (Reachable k-Connected Centers):

4

Let P be a given set of sensors and K be a reachability matrix. Let $C = \{c_1, \cdots, c_k\}$ be a set of centers, each has a maximum velocity $v_C > 0$. A *reachable k-connected center* from C is a set $C^* = \{c_1^*, \cdots, c_k^*\}$ where $\|c_i - c_i^*\| \leq v_C$ for every $i = 1, \cdots, k$. Moreover,

$$C^* \in \arg\min_{C',\delta}(rb(P,C') + \delta) \text{ s.t.} \tag{14}$$

$$(c-p)^T K(c-p) \leq \delta, \ \forall c \in C', \ \forall p \in \mathcal{N}(c).$$

Here, K is the reachability matrix from (13), and $\mathcal{N}(c)$ is the communication neighborhood from Definition 3.2. A solution with $\delta \leq 1$ indicates that a reachable configuration is feasible.

From the matrix K we can derive the *expiration time* or time, in seconds, for which we are guaranteed connectivity between a center c and its connectivity neighborhood:

$$t_e = \max\left(\min_{c \in C, p \in \mathcal{N}(c)} \frac{r_\chi - \|c-p\|}{v}, 0\right). \tag{15}$$

Here $v = \{v_C, v_S\}$ where v_C is used if the vehicle p is a neighboring center so that $p = T(c)$, and v_S is used if p is a sensor so that $p \in P$. Also, $r_\chi = \frac{1}{\sqrt{\lambda}}$ and λ is the largest eigenvalue of K.

Lemma 3.4: Given a set of sensors with positions $P \subseteq \mathbb{R}^d$ and maximum velocity bound $v_S \in \mathbb{R}$, a communication radius R, and a velocity limitation $v_C \in \mathbb{R}$ for each center, we have that centers with positions $C \subseteq \mathbb{R}^d$ will maintain a connected configuration (Definition 2.2) with P over t_e seconds for any sensor motion satisfying (2) if the convex condition

$$(c_i - p)^T K(c_i - p) \leq 1, \ \forall c_i \in C, \ \forall p \in \mathcal{N}(c_i) \tag{16}$$

holds. Where t_e is the expiration time given by Equation (15), the matrix K is positive definite, of appropriate dimension, depends on v_S, v_C and R, and is calculated for our problem in (13), and the connectivity neighborhood $\mathcal{N}(c_i)$ is from Defintion 3.2.

Proof: The condition (16) is calculated using ellipsoidal approximation methods from [17]. The collection of connectivity neighborhoods $\mathcal{N}(c_i)$ for all $c_i \in C$ is guaranteed to cover all sensors and connect all centers by the k-connected center formulation. Therefore sets P and C satisfying Equation (16) maintain a connected state over the entire network since this implies that $(p - c_i) \in \bar{X} \subset X$ for all c_i, which is a sufficient condition for connectivity by the description of the sets (11) and (12). The amount of time (in seconds) that a connected configuration is maintained for C is given by the expiration time t_e that is computed from K. The largest eigenvalue of K gives us a lower bound on the radius r_{χ_i} of the reachability set for c_i, and thus any relative state $(p - c_i)$ with magnitude $\|(p - c_i)\| \leq r_{\chi_i}$ is guaranteed to remain connected, where p is in the connectivity neighborhood $\mathcal{N}(c_i)$ of c_i. The minimum value of t_e for a given C,P,v_S,v_C is given by (15). ∎

C. Exact Algorithm for Reachable k-Connected Centers

We combine the results from the previous two subsections on k-connected centers and reachability analysis to provide an algorithm for returning communication vehicle placements that minimize the reachable k-connected centers cost from (14). This solution is optimal for the k-connected center problem and satisfies a sufficient condition for reachability if $\delta^* < 1$ (see Algorithm 1). Algorithm 1 can also be easily altered to only return solutions that satisfy reachability. Since the reachability condition from (16) involves an inner ellipsoidal approximation for computational tractability, we point out that there may exist configurations C^* for P that do not satisfy (16) but are reachable.

Suppose that we are given the partition of the input set P into k clusters (one cluster per center in C^*), and the pairs from the set $1, \cdots, k$ that correspond to the edges of a minimum spanning tree of C^*. Then, a k-connected center set C^* can be computed in polynomial time using a convex program as in Algorithm 1. Using exhaustive search on all candidate solutions, we can thus compute C^* in $n^{O(k)}$ time.

---

input : Set P of sensors, set $C = \{c_1, \cdots, c_k\}$ of k centers, reachability matrix $K \in \mathbb{R}^{d \times d}$ from (13), v_S and v_C from (2) and (4)
output: Reachable k-connected center C^* from C, expiration time t_e, and reachability parameter δ^*

for *Every spanning tree T over k nodes* **do**
  for *Every partition $\mathcal{N}(c_1), \cdots, \mathcal{N}(c_k)$ of P* **do**
    $(C', r, \delta) \in \arg\min_{C' = \{c_1', \cdots, c_k'\}, r, \delta} r + \delta$ s.t. :
    $\|c - q\| \leq r, \ \forall q \in \mathcal{N}(c), \ \forall c \in C'$
    $\|c_i' - c_i\| \leq v_C, \ i \in \{1, \ldots, k\}$
    $(c - q)^T K(c - q) \leq \delta, \ \forall q \in \mathcal{N}(c) \ \forall c \in C'$
    if $r < r^*$ **then**
      $r^* = r; \ C^* = C; \ \delta^* = \delta$
    end
  end
end
$t_e \leftarrow \max\left(\min_{c \in C, p \in \mathcal{N}(c)} \frac{r_\chi - \|c-p\|}{v}, 0\right),$
$v \in \{v_S, v_C\}$
return: r^*, C^*, t_e, δ^*

Algorithm 1: Exact Algorithm for k-connected centers with Reachability

---

Theorem 3.5: Let P be a set of points in $\mathbb{R}^d$ and $k \geq 1$ an integer. Algorithm 1 provides an exact solution to the reachable k-connected centers problem from Definition 3.3, where the resulting set of centers C^* is the set of centers of cardinality $|C^*| = k$ that minimizes the

cost $rb(P, C)$ over every set C of centers that satisfy the constraints . Moreover, the configuration C^* guarantees connectivity over the entire heterogeneous network for a minimum of t_e timesteps where t_e is the expiration time for C^* and is defined in (15). The algorithm runs in $n^{O(k)}$ time.

Proof: The proof for the optimality of the solution C^* under the constraints follows from the exhaustive search enumeration for the exact algorithm we presented for k-connected center. It follows from Lemma 3.4 that the configuration C^* guarantees connectivity over t_e timesteps. The Q loop in Algorithm 1 runs computations over subsets of P of size $\binom{n}{d+1}$ for each of the k centers, and the inner loop performs computations over all spanning trees for a graph with k nodes. Since convex quadratically constrained quadratic programs can be solved in polynomial time and we assume $n \geq k$ we have that the dominating complexity is $n^{O(k)}$. ∎

IV. APPROXIMATIONS

A. Tracking a Representative Set of Sensors

For large numbers of sensors n, computing the exact solution for the k-connected centers can consume a prohibitive amount of time. It would be more desirable to instead compute the exact solution over a carefully chosen subset of sensors in a way that the induced approximation error can be bounded.

Theorem 4.1: For every $\alpha \geq 1$, an α-approximation to the k-center of a set $P \subseteq \mathbb{R}^d$ is an $(3 + 2\alpha)$-approximation to the k-connected center of P.

Proof: (sketch) Let C^* be a connected k-center of P, and let C be an α-approximation for the k-center of P. By definition, $rb(P, C) = \max\{r(P, C), b(C)\}$, $r(P, C^*) \leq rb(P, C^*)$, and $r(P, C) \leq \alpha r(P, C^*)$. Combining the last inequalities yields

$$r(P, C) \leq \alpha r(P, C^*) \leq \alpha rb(P, C^*). \quad (17)$$

Hence,

$$rb(P, C) \leq \max\{\alpha rb(P, C^*), b(C)\}. \quad (18)$$

It is left to bound $b(C)$. For every $c \in C$, define $f(c) \in C^*$ to be the closest center to c in C^*. For every $c \in C^*$, define $g(c) \in C$ to be the closest center to c in C. Let $G(C^*, T^*)$ be an MBST of C^*. Let $T = T_1 \cup T_2$, where

$$T_1 = \left\{ \big(g(c), g(c')\big) \mid (c, c') \in T^* \right\}, \text{ and}$$

$$T_2 = \left\{ \big(c, g(f(c))\big) \mid c \in C \right\} \cup \left\{ \big(g(f(c)), c\big) \mid c \in C \right\}.$$

Since $G(C^*, T^*)$ is a spanning tree,

$$b(C) \leq \max_{(c, c') \in T} \text{dist}(c, c'). \quad (19)$$

Suppose that $(c, c') \in T$. It is not hard to verify that

$$\text{dist}(c, c') \leq (3 + 2\alpha) r(P, C^*) \quad (20)$$

by case analysis: (i) $(c, c') \in T_1$ and (ii) $(c, c') \in T_2$.

Together with (19), we obtain $b(C) \leq (3 + 2\alpha)r(P, C^*)$. Plugging the last inequality in (18) proves

the theorem, as

$$\begin{aligned} rb(P, C) &\leq \max\{\alpha rb(P, C^*), (3 + 2\alpha)rb(P, C^*)\} \\ &\leq (3 + 2\alpha)rb(P, C^*). \end{aligned}$$

∎

B. Coresets

To produce gains in efficiency we use a data structure called a coreset for the k-center problem and prove the same coreset has the desired properties for the k-connected and reachable k-connected center problems.

Hence, running our exact Algorithm 1 on the coreset S instead of P, would yield a $(1 + \varepsilon)$-approximation to the k-centers of P. The corresponding running time would be reduced then from $n^{O(k)}$ to $|S|^{O(k)}$. Clearly, $S = P$ is a coreset for P. However, if we can compute a coreset of size $|S| \ll n$ the running time on the coreset would be significantly smaller.

---

input : A set $P \subseteq \mathbb{R}^d$ of n robots, $k \geq 1$ centers, and a constant $\varepsilon > 0$
output: A (k, ε)-coreset S of size $O(\frac{k}{\varepsilon})$ for P

1 $i \leftarrow 0$; $P_0 \leftarrow P$;
2 **while** $|P_i| > k$ **do**
3   Pick a random set T_i of k robots from P_i ;
4   Remove half of the closest robots $Q_i \subseteq P_i$ to T_i ;
    `/* Continue recursively with`
       `the remaining robots.   */`
5     $P_{i+1} \leftarrow P_i \setminus Q_i$;
6     $i \leftarrow i + 1$;
end
7 $T \leftarrow T_0 \cup \cdots \cup T_{i-1} \cup T_i$;
8 **for** *each $p \in T$* **do**
9   Construct a d-dimensional grid G_p of side length $(\varepsilon/\sqrt{d}) \cdot \text{Dist}(P, T)$ that is centered
10 at p Pick an arbitrary representative robot $q \in P$ from every non-empty cell of G_p
end
11 $S \leftarrow$ the union of representatives that were selected at Line 8;
12 Return S

Algorithm 2: A (k, ε)-coreset S for P

Definition 4.2 ((k, ε)-coreset for k-center problem): Let P be a set of points in $\mathbb{R}^d$, $k \geq 1$ be an integer, and $\varepsilon > 0$ be a constant. A set $S \subseteq P$ is called a (k, ε)-*coreset* for P if for every given set of k centers C in $\mathbb{R}^d$, we have

$$r(S, C) \leq r(P, C) \leq (1 + \varepsilon)r(S, C). \quad (21)$$

A (k, ε)-coreset S for the k-center problem is also a (k, ε)-coreset for the reachable k-connected center problem so that for any C in $\mathbb{R}^d$ where $|C| = k$ it

holds that

$$rb(S,C) \leq rb(P,C) \leq (1+\varepsilon)rb(S,C). \quad (22)$$

This follows from (21) since the bottleneck edge is only a function of C and not the input set P and $rb(P,C) = \max\{r(P,C), b(C)\}$. Since configurations $\tilde{C}$ that satisfy the reachability condition from (16) are a subset of C, and Equation (21) holds for all C then this property also holds for any constrained sets $\tilde{C}$. Algorithm 2 returns a coreset S as stated in the following theorem whose proof can be found in [14]:

Theorem 4.3: Let $S \subseteq P$ be a set of points that is returned by a call to Algorithm 2 with $P \subseteq \mathbb{R}^d$, $k \geq 1$ and $\varepsilon > 0$. Then, with high probability, S is a (k, ε)-coreset for P. That is, for every set C of k centers in $\mathbb{R}^d$ we have

$$rb(S,C) \leq rb(P,C) \leq (1+\varepsilon)rb(S,C).$$

The running time of the algorithm is $O(nk)$.

Since we have that the set S returned by Algorithm 2 is a (k, ε)-coreset for P we have that $rb(S,\tilde{C}) \leq (1+\varepsilon)rb(P,C^*)$ where $\tilde{C}$ is the reachable k-connected center computed over the coreset S and C^* is the reachable k-connected center computed over the entire input set P since

$$rb(S,\tilde{C}) \leq rb(P,\tilde{C}) \leq (1+\varepsilon)rb(S,\tilde{C})$$
$$\leq (1+\varepsilon)rb(S,C^*) \leq (1+\varepsilon)rb(P,C^*).$$

This formalized in the following corollary:

Corollary 4.4: Let P be a set of n points and let $\varepsilon > 0$ be a constant. Then a $(1+\varepsilon)$-approximation for the reachable k-connected center of P can be computed in $O(nk) + (\frac{k}{\varepsilon})^{O(k)}$ time.

Empirical results showing the reduction in the approximation error with size of the coreset is shown in Figure 4.

Overview of Algorithm 2. We pick a small random sample T_1 from P. Such a random sample has the property that it "hits" large clusters Q_1 of robots, but probably misses outliers. Hence, in Line 4 we remove only the half closest robots to T_1, which are approximated well, and keep the remaining robots. We continue recursively until no robots are left. This yields $O(\log n)$ iterations that corresponds to $O(\log n)$ sample sets. Since $T \subseteq P$ we have that $r(S,C^*) \leq r(P,C^*)$ for the k-center of P. In Lines 8–11 we turn this $O(1)$-factor approximation into $O(\varepsilon)$-approximation by constructing a grid G_p around every robot p in S. The distance between two points that are in the same cell of the grid is at most $\varepsilon rb(P,C^*)$

Corollary 4.5: Let P be a set of n robots locations in $\mathbb{R}^d$, $k \geq 1$ be an integer, and $\varepsilon > 0$ be a constant. Let S be the output of Algorithm 2, and let $\tilde{C}$ be the reachable k-center of S computed by Algorithm 1. Then $\tilde{C}$ is a $(1+\varepsilon)$-approximation to the reachable k-center of P.
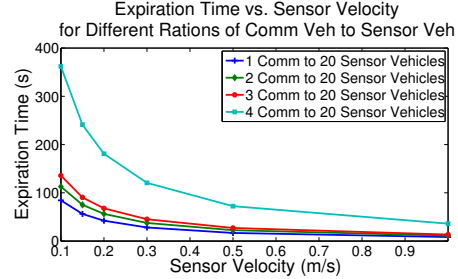

Fig. 2: Effect of sensor velocity on expiration time studied for a fixed $R = 60$m and communication (router) vehicle speed $v_C = 1.5m/s$. Different curves correspond to different ratios of comm vehicles to sensors k/n and the slope of the curves become shallower as the number of router vehicles k increases per unit area.

V. RESULTS

In this section we provide empirical results that study i) the effect of sensor velocity and ratio of centers k to sensors n on the expiration time t_e (see Figure 2), ii) computation vs. expiration time and how the expiration time can be used as a guide for switching between exact and approximate methods (see Figure 3), and iii) the approximation error induced by using a representative set S vs. the size of this representative set (see Figure 4).

We examine two sensor behaviors, where 1) sensors are distributed uniformly over a fixed area but vary their maximum velocities and 2) sensors begin at the center of the environment and move outwards radially at a speed of $1m/s$ permitting worst-case analysis. For the first case we investigate the effect of increasing sensor velocities on the *expiration time*, or minimum bound on the time that the centers generated by Algorithm 1 are guaranteed to maintain communication with the sensors. It must hold that the maximum allowable velocity of the centers is at least that of the sensors in order to maintain connectivity. We also vary the ratio of sensors to communication vehicles in order to demonstrate how a growing density of communication vehicles increases the attainable expiration times.

The second case assumes a fixed number of 2 centers that must provide connectivity for 5 sensors that are moving radially outward from the environment center. Figure 3 demonstrates the motivation for switching from an exact optimization method, to an approximate method that uses a coreset. We perform this switch when the computation time of the exact algorithm from Algorithm 1 reaches the expiration time and thus the center locations must be updated more quickly in order to maintain connectivity. We show that performing the exact algorithm over a coreset of size $n/2$ improves our computation time 12-fold with an average approximation cost of $1.13 \cdot OPT$. As the sensor vehicles move farther outwards towards the communication limits of the centers we can switch to a $(3 + 2\alpha)$-approximation algorithm which computes a k-connected center solution
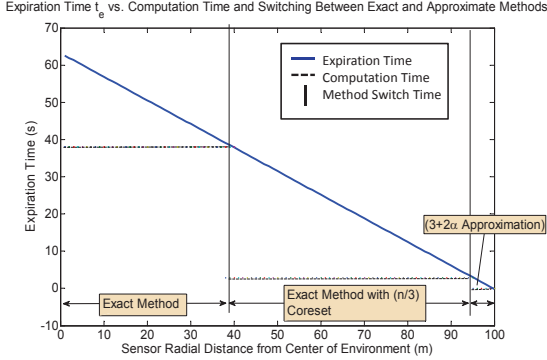
exponential, even for small input sets the computational savings is significant using coresets.

VI. CONCLUSION

In this work we define a formal framework, design provable algorithms, and provide empirical case studies for the problem of providing communication coverage to mobile robots that are moving over unknown trajectories. We develop a reachable k-connected centers approach and return a solution of router vehicle placements that maximize the amount of time over which connectivity is guaranteed regardless of sensor movement during that time.

Fig. 3: Plot of t_e vs. radial sensor distance from center of environment. Computation time (horizontal dashed lines) is compared against expiration time (solid blue line) and switch times (solid vertical lines) between exact and approximate methods is demonstrated.
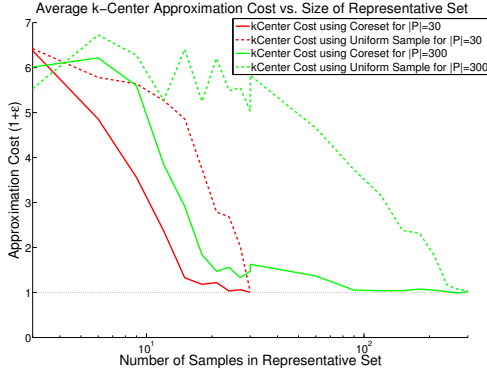

Fig. 4: This plot shows aggregate results over 1000 runs for the error induced by using a representative set of size $|S|$ for the input set P vs. increasing representative set size. The plot shows that a *coreset* (solid line) provides better performance with approximation error $\varepsilon \leq 0.34$ vs $\varepsilon = 3.8$ for a uniform random sample (dashed line) for only $|P|/2$ sample points when $|P| = 30$, and $\varepsilon \leq 0.05$ vs. $\varepsilon = 2.7$ for a uniform random sample with less than $|P|/3$ sample points when $|P| = 300$.

in 0.01 seconds but at an approximation cost of $(3+2\alpha)$.

Figure 4 shows the improvement in the induced error, ε, of using a coreset as a representative set for P vs. the size of the representative set. We compute a 2-approximation to the k-center cost on different sized input sets, $|P| = 30$ and $|P| = 300$. This plot shows the ratio of this k-center cost computed over a representative set to the k-center cost computed over the entire input set P. We contrast the performance of using a uniform random sample (dashed line) to that of using a coreset (solid line) and show that the coreset provides better performance with approximation error $\varepsilon \leq 0.34$ for only $n/2$ sample points for $n = 30$ and $\varepsilon \leq 0.05$ for less than $n/3$ sample points when $n = 300$. Since the computation time for computing the exact k-center cost is

REFERENCES

[1] P. Agarwal, H. Edelsbrunner, O. Schwarzkopf, and E. Welsl, "Euclidean minimum spanning trees and bichromatic closest pairs," *Discrete and Computational Geometry*, vol. 6, no. 1, 1991.

[2] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," ../../cvx, Apr. 2011.

[3] M. Zavlanos, M. Egerstedt, and G. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525 –1540, sept. 2011.

[4] A. Cornejo, F. Kuhn, R. Ley-Wild, and N. Lynch, "Keeping mobile robot swarms connected," in *Proceedings of the 23rd international conference on Distributed computing*, ser. DISC'09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 496–511.

[5] N.Michael, M. M. Zavlanos, V. Kumar, and G.Pappas, "Maintaining connectivity in mobile robot networks," *Experimental Robotics*, pp. 117–126, 2009.

[6] M. Schuresko and J. Cortes, "Distributed tree rearrangements for reachability and robust connectivity," *Intell. Robot. Syst.*, vol. 56, no. 1-2, pp. 99–126, sept. 2009.

[7] O. Tekdas, W. Yang, and V. Isler, "Robotic routers: Algorithms and implementation," *Int. Journal of Robotics Research*, vol. 29, no. 1, 2010.

[8] S. Gil, M. Schwager, B. Julian, and D. Rus, "Optimizing communication in air-ground robot networks using decentralized control," in *Proc. ICRA*, 2010.

[9] S. Gil, S. Prentice, N. Roy, and D. Rus, "Decentralized control for optimizing communication with infeasible regions," in *Proceedings of the 15th International Symposium on Robotics Research*, 2011.

[10] K. Savla, G. Notarstefano, and F. Bullo, "Maintaining limited-range connectivity among second-order agents," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, 2009.

[11] T. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, 1985.

[12] P. Agarwal, S. Har-Peled, and K. Varadarajan, "Geometric approximation via coresets," *Discrete and Computational Geometry*, vol. 52, 2005.

[13] S. Har-peled and S. Mazumdar, "Coresets for k-means and k-median clustering and their applications," in *In Proc. 36th Annu. ACM Sympos. Theory Comput*, 2003, pp. 291–300.

[14] D. Feldman, A. Fiat, M. Sharir, and D. Segev, "Bi-criteria linear-time approximations for generalized k-mean/median/center." in *Symposium on Computational Geometry'07*, 2007, pp. 19–26.

[15] J. Ding, E. Li, H. Huang, and C. J. Tomlin, "Reachability-based synthesis of feedback policies for motion planning under bounded disturbances," in *ICRA*, 2011, pp. 2160–2165.

[16] J. H. Gillula, G. Hoffmann, H. Huang, M. P. Vitus, and C. J. Tomlin, "Applications of hybrid reachability analysis to robotic aerial vehicles," *I. J. Robotic Res.*, vol. 30, no. 3, pp. 335–354, 2011.

[17] D. Bertsekas, "Control of uncertain systems with a set-membership description of the uncertainty," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 1971.