Relay Vehicle Formations for Optimizing Communication Quality in Robot Networks

Md Mahbubur Rahman[1], Leonardo Bobadilla[1], Franklin Abodo[1], Brian Rapp[2]

Abstract— **In this paper, we solve the problem of relay robot placement in multi-robot missions to establish or enhance communication between a static operator and a number of remote units in an environment with known obstacles. We study the hardness of two different relay placement problems: 1) a chain formation of multiple relay robots to transmit information from an operator to a single unit; and 2) a spanning tree of relays connecting multiple remote units to the operator. We first build a communication map data structure from a layered graph that contains the positions of the relays as the unit moves. This structure is computed once and reused throughout the mission, significantly reducing plan re-computation time when compared to the best-known solution in the literature. Second, we create a min-*arborescence* tree that forms a connected component among the operator, relays, and units, and that has an optimal communication cost. Finally, we validate our ideas through software simulations, hardware experiments, and a comparison of our approach to state-of-the-art methods.**

I. Introduction

The use of robotic systems, whether autonomous or remotely-operated, presents the opportunity to accomplish missions with reduced cost and risk to humans. These missions may include surveillance, rescue, or disaster response. The ability to communicate with remote robotic systems can be compromised, however, by distance, obstacles in the environment, or electromagnetic interference (intentional or natural). In an underground tunnel, building corridor, for example, communication via satellite is unlikely to be successful. To alleviate these difficulties, communication relays may be established, and these relays can also be mounted on robotic systems.

Relay-based communication has practical applications in scenarios where traditional communication systems are compromised or broken. Such scenarios can be found in disaster areas, military operations, and other locations where either the traditional communication is absent or a manned mission is not safe. In these communication-constrained environments, one or more unmanned units can be used to collect data, monitor activity, or take other actions. These robots are remotely controlled by an operator who stays in a safe region. However, due to obstacles and terrain the signal degrades or drops over long distances and we need to deploy intermediate relay robots in between the operator and the remote units in order to maximize communication quality. Example scenarios for this problem are shown in Figure 1(a), where we need to build a relay chain to serve a single remote unit, and in Figure 1(b), where we need to construct a spanning tree to serve three remote units. As the number of relays is limited, an optimal placement plan is required to achieve the best communication signal possible to the remote units using the available relays.

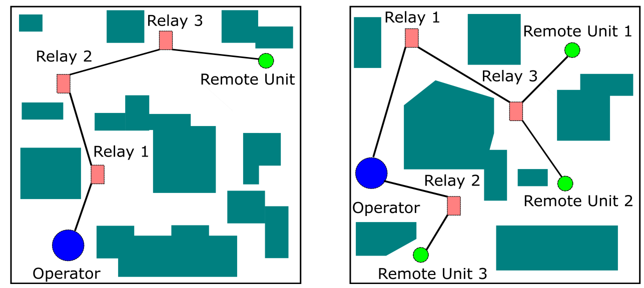


Fig. 1. (a) A chain consisting of three robots that relay communication from an operator to a remote unit; (b) A minimum spanning tree incorporating three relays, optimizing communication from an operator to three units.

Contributions: In this paper, we investigate how to connect one or more remote units to a base controller through a limited number of intermediate relay robots in a communication constrained-environment. We study the computational complexity of a set of different problems and propose methodologies that can create a relay chain or tree as required by the scenario. We proved that this family of problems is intractable and propose resolution complete algorithms that use a decomposition of free space into a finite number of cells (such as a grid or a mesh). We design a highly adaptive and reusable data structure as a communication map that is used to extract the positions for a given number of intermediate relays. This map is computed only once for a fixed operator, and the positions of a given number of intermediate relays can be extracted for different positions of the unit as it moves. Therefore, we eliminate the re-computation of the entire deployment each time either the unit moves or the number of relays changes; a significant reduction of computation compared to the methods used in [1] and [2]. In the cases of serving multiple units, a limited branching Steiner tree [3] is adapted for our problem. We generate an optimal min-*arborescence* tree [4] that serves as a map of positions for multiple relays to serve multiple units.

II. Related Work

The robotic relay placement problem has been studied in the literature with a focus on controlling the robots and the formation of a relay chain. The best known solution for our first problem about the robotic relay chain formation (Figure 1(a)) was proposed in [1], [2]. Two different algorithms, a modified Bellman-Ford algorithm [5] and a dual ascent algorithm, were

[1]M. Rahman, F. Abodo and L. Bobadilla are with the School of Computing and Information Sciences, Florida International University, Miami, FL, 33199, USA. mrahm025@fiu.edu, fabod001@fiu.edu, bobadilla@cs.fiu.edu

[2]B. Rapp is with the Army Research Lab, USA. brian.m.rapp2.civ@mail.mil

used on a grid to find the shortest sequence of grid points for placing the given number of robot relays. Although their solution is able to form a relay chain, frequent re-computation of the chain is required each time either the unit moves to a new location or the number of relays changes. In contrast, we develop a reusable data structure as a static placement map that is computed once and used to extract the new locations of the available relays when the unit moves throughout the mission. Thus, our solution eliminates significant re-computation and re-planning time in a mobile robotic system.

Our second problem, multi-unit multi-relay tree formation, is connected to the limited branching *Steiner* tree discussed in [3]. Although the general problem is known to be NP-Hard, the authors proved that a polynomial time algorithm can compute a tree for a fixed number of branching and terminal nodes. We have adapted the ideas in [3] for our problem and expanded them for implementation purposes. Another relay formation solution using Markov chains is proposed in [6] where the relays move based on the inputs from their neighbors. However, obstacles were not considered, and the robots did not form other topologies besides a chain.

This research is closely related to wireless sensor networks, mesh networks, and multi-hop dynamic wireless networks [7]. However, most of the solutions are related to area coverages for which static relay nodes are used that are not capable of adjusting their locations through movement. Our ideas are naturally connected to visibility graph-based [8] planning and art gallery problems [9], [10] that guard polygons through visibility. However, the solution is a minimum number of nodes required to observe the whole galley, which is not applicable in our problem where we need to achieve the best communication using the given number of nodes.

Our work also has similarities to leader-follower robot formation [11], [12], [13] where a number of robots position themselves according to the policy distributed by their leader. Although a consensus-based control algorithm is provided in [11] and a dynamic controller was designed in [13], no obstacles are considered in either work. A visual odometry is used in [12] to keep the leader in sight, but the calculated trajectories and positions do not guarantee any optimality.

III. PRELIMINARIES

We will consider a two-dimensional *environment* $\mathcal{W} = \mathbb{R}^2$ that is filled with polygonal *obstacles* $\mathcal{O}$ as illustrated in Figure 1. In this environment, there is a set of m *relay vehicles* $A_1, A_2, \ldots, A_m$ and p remote units $B_1, B_2, \ldots, B_p$ that need to be connected to a static *operator* S. We define the collision-free space as $\mathcal{W}' = \mathcal{W} \setminus \mathcal{O}$ where the units and relays present in the world can move freely. The remote units are modeled as point robots and a unit B_j has configuration space $\mathcal{B}_j$, where a particular configuration $r_j \in \mathcal{B}_j$ is defined as $r_j = (x, y) \in \mathcal{W}'$. Similarly, the configuration space for the operator S is defined as $\mathcal{S}$, where an operator's position $s \in \mathcal{S}$ is denoted by $s = (x, y)$. The relay vehicles are modeled as car-like robots and a particular vehicle A_i has a configuration space $\mathcal{C}_i$ and the positions $q_i \in \mathcal{C}_i$ are defined as $q_i = (x, y, \theta) \in \mathcal{W}' \times [0, 2\pi)$ [14].

A. Communication Quality

For any two points on the plane $\rho_1, \rho_2 \in \mathcal{W}'$, the received power is inversely proportional to, $d^{\delta}(\rho_1, \rho_2)$ [15], [16], where $d(\rho_1, \rho_2)$ is the distance between ρ_1 and ρ_2 and δ is the loss coefficient. δ depends on the nature of the workspace and is generally set to 2 when the sender (placed at ρ_1) and receiver (placed at ρ_2) have free Line of Sight (LoS) (see Table 4.1 in [16]). Therefore, the free space communication cost, f_F increases with the increase of distance i.e, proportional to the quadratic distance, d^2, and is defined as,

$$f_F(\rho_1, \rho_2) = \begin{cases} \gamma d^2(\rho_1, \rho_2) & \text{if } d(\rho_1, \rho_2) < d_{th} \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

Here γ is a constant that depends on the transmitter [15] and d_{th} is the distance threshold beyond which no communication can be established. Communication signal is further attenuated by diffraction, fading and and/or multipath propagation effects due to the presence of obstacles between the sender and receiver [16].

Let the path loss in the presence of obstacles and terrain be $f_{\mathcal{O}}(\rho_1, \rho_2, \mathcal{O})$, which includes the costs resulting from diffraction (f_{DF}), fading (f_{FA}), and/or multipath propagation.

$$f_{\mathcal{O}}(\rho_1, \rho_2, \mathcal{O}) = \begin{cases} 0 & \text{if } \overline{\rho_1 \rho_2} \text{ has LoS} \\ f_{DF}(\mathcal{O}) + f_{FA}(\mathcal{O}) & \text{otherwise} \end{cases} \tag{2}$$

Finally, the total communication cost f_C between ρ_1 and ρ_2 is defined as:

$$f_C(\rho_1, \rho_2) = f_F(\rho_1, \rho_2) + f_{\mathcal{O}}(\rho_1, \rho_2, \mathcal{O}) \tag{3}$$

B. Relay Placement Problems

Our first problem of interest is to develop a solution for the relay placement problem involving an operator, a number of relay robots, and a remote unit. Given the operator's position s and a remote unit's position r, we need to calculate a set of relay robots' positions $q_1, q_2, \ldots, q_m$ such that they form a communication chain. The communication cost is defined as,

$$f_C^L = f_C(s, q_1) + \sum_{1 \le i < m} f_C(q_i, q_{i+1}) + f_C(q_m, r) \tag{4}$$

We are required to solve the problem of creating a reusable placement map that gives the best placements for a given number of relay vehicles. Therefore, we define a communication map corresponding to the static operator s as $M_c^s : \mathcal{B} \rightarrow \mathcal{C}^n$. Accordingly, our first problem is:

Problem 1: MULTI-RELAY CHAIN - Finding Optimal Positioning of a Set of Relay Robots on a Chain. *Given the fixed positions r and s corresponding to a unit B and an operator S, find m points $q_1, \ldots, q_m$ corresponding to the m relay vehicles $A_1, A_2, \ldots, A_m$ in the free space that form an $m+1$-link m hop path to connect s to r and minimize f_C^L.*

We extend the multiple-relay single-unit problem to a multiple-relay multiple-unit problem. Consequently, we have p unit positions $r_1, \ldots, r_p$ that must be connected to s through m relays. Therefore we define our second problem as a MULTI-RELAY MULTI-UNIT problem:

Problem 2: MULTI-RELAY MULTI-UNIT - Finding Optimal Positioning of a Set of Relay Robots That Serve a Number of Remote Units.
Given a set of fixed positions $r_1, \ldots, r_p$ *of* p *units and the position* s *of one operator, compute the optimal positions* $q_1, q_2, \ldots, q_m$ *of* m *relay robots on the plane that form a connected component among the operator, relays and units while the term* $\sum_{1 \le i \le p} \min_{1 \le j \le m} f_C(r_i, q_j) + \min_{1 \le j \le m} f_C(s, q_j)$ *is minimized.*

In this case, the optimal solution is a tree $T = (V, E)$ that spans over the operator, p remote unit positions, and m available relay positions. Accordingly, the communication cost to be minimized of this multi-unit system is defined as:

$$f_C^T(T) = \sum_{(u,v) \in E} [\alpha_1 f_C(u, v) + \alpha_2(deg(u) + deg(v))] \quad (5)$$

where, α_1 $\alpha_2 \in [0, 1]$ are weighting factors such that $\alpha_1 + \alpha_2 = 1$. The terms $deg(.)$ denotes the degree of a node and it is used to constrain the number of flows/connections per node.

IV. Methods

A. Single Unit Multiple Relay Placement

A MULTI-RELAY CHAIN problem is shown in Figure 1(a) where we want to form a relay chain between the operator and the remote unit. However, the problem becomes NP-Hard on a plane filled with obstacles as stated below.

Proposition 4.1: The MULTI-RELAY CHAIN problem in a polygon with holes is NP-Hard.

Proof: (Sketch) Our problem is similar to the shortest $m+1$-link paths in polygons with holes as discussed in [17], [18]. The bi-criteria shortest path decision problem was proven to be NP-Complete [17] when we need to decide if a path with $m + 1$ links is the shortest. Therefore, the optimization version of calculating the shortest $m+1$ link path (our MULTI-RELAY CHAIN problem) is NP-Hard. Although we use a communication cost metric f_C, it is a function of the distance metric d and does not reduce the hardness of the problem. ■

Therefore, we employ discretization as shown in Figure 2(a)-(b) instead of solving the problem in the continuous plane. We convert the world $\mathcal{W}' = \mathcal{W} \setminus \mathcal{O}$ into a grid (such as a *Sukharev* grid [19]) with n grid points $\Omega = \{g_1, g_2, \ldots, g_n\}$. An example environment grid Ω is shown in Figure 2(a) where the operator S stays in cell 0. A graph representation $G(V, E)$ of Ω, based on communication cost f_C, is drawn in Figure 2(b) where the node set V is composed of all the grid points that are not inside the obstacles $\mathcal{O}$, and is defined as $V = \{v_i | v_i \equiv g_i \in \Omega \text{ and } g_i \notin \mathcal{O}\}$. Here, a node $v_i \in V$ is equivalent to a grid point $g_i \in \Omega$, but contains additional attributes such as identifier, cost, neighbors, and parent. Each node $v \in V$ has a unique identifier $v.id$ that is used to identify the node. The set of undirected edges E is defined as $E = \{(u, v) : f_C(u, v) < \infty\}$ where the weight of an edge is computed by (3). For illustration purposes, initially the communication between two grid points is blocked by the obstacles. However, we will show other general cases in the experimental section where the signal is allowed to penetrate the obstacles.

Next, we compute the communication map M_c^s using Algorithm 1. A layered directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $m+2$ levels $l_0, l_1, \ldots, l_{m+1}$ for m available relay robots is computed (see Figure 2(c)) based on the original graph G. Level l_0 contains only one node $v_s^0 \equiv v_s \in V$ corresponding to the static operator's position s which also represents the root of the tree. Each of the subsequent layers l_i, where $1 \le i \le m+1$, will copy all the nodes $V \setminus v_s$ of the original graph G. This means a particular layer l_i contains the nodes $\mathcal{V}_i = \{v_1^i, v_2^i, \ldots, v_{|V|}^i\}$ and, for a node $v_k^i \in V_i$, the identifier $v_k^i.id = v_k.id$, where $v_k \in V$ is the corresponding original node in G. Additionally, the nodes at different layers with the same index have the same identifier, which means $v_k^1.id = v_k^2.id = \cdots = v_k^{m+1}.id$ (see Figure 2(c)). Finally, the node set $\mathcal{V}$ for the graph $\mathcal{G}$ is defined as,

$$\mathcal{V} = \mathcal{V}_0 \cup \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_{m+1} \quad (6)$$

which contains $O((m + 1) \cdot |V|)$ nodes. A directed edge $(u, v) \in \mathcal{E}$ is allowed only between the nodes of any two consecutive layers l_i and l_{i+1} (lexicographic order) if and only if $(u', v') \in E$ (which means $f_C(u, v) < \infty$) where $u.id = u'.id$ and $v.id = v'.id$ for $u', v' \in G.V$:

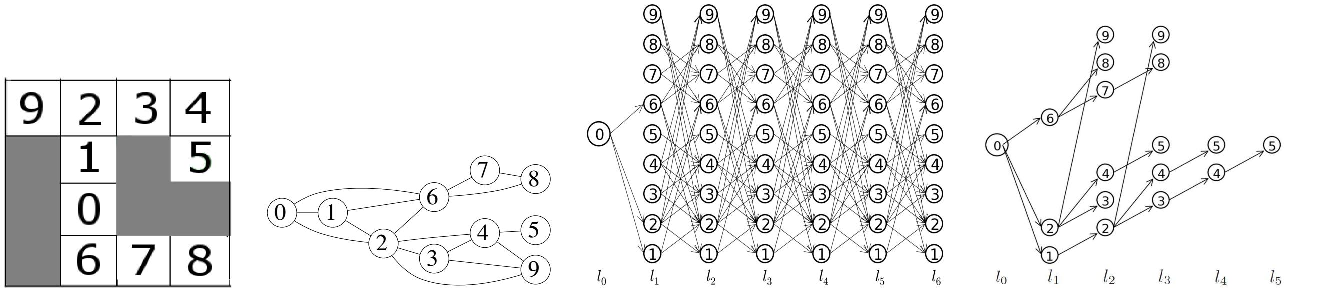
$$\mathcal{E} = \{(u, v) : u \in l_i, v \in l_{i+1} \text{ and } f_C(u, v) < \infty \ ; \ i \le m\} \quad (7)$$

Once the layered graph $\mathcal{G}$ is constructed, we compute a modified shortest path tree that results in our communication map M_c^s. The resulting tree is constructed by exploring $\mathcal{G}$ layer-wise in a lexicographic order while removing the unnecessary nodes that have already attained optimality. Therefore, we modify the breadth first graph search (BFS) [5] algorithm to explore layer by layer and compute the shortest chain from the root v_s to each of the nodes. Line 3 of Algorithm 1 initializes the exploration by enqueuing v_s into a queue Q. In order to compute the shortest path tree, we introduce a hash table h of length $|V|$ that uses $v.id$ as the keys and is initialized to ∞ (line 4). We defined earlier that a particular node $v \in \mathcal{V}$ has the same key $v.id$ in all the layers of $\mathcal{G}$ where its instances appear (see the numbering in Figure 2(c)). Therefore, h is used to keep track of the lowest cost of each node $v \in V$ of the original graph G as we explore throughout the levels of $\mathcal{G}$.

Algorithm 1 multiRelaySingleUnit($G(V, E)$)

```
1: 𝒢(𝒱, ℰ) = calculateGraph(G)
2: v_s.cost = 0, and v.parent = NULL; ∀v ∈ 𝒱
3: Enqueue(Q, v_s)
4: h[v.id] = ∞; ∀v ∈ G.V
5: while Q ≠ ∅ do
6:     u = Dequeue(Q)
7:     for v ∈ u.Neighbors do
8:         if u.cost + f_C(u, v) < h[v.id] then
9:             v.parent = u
10:            v.cost = u.cost + f_C(u, v)
11:            h[v.id] = v.cost
12:            Enqueue(Q, v)
13:        end if
14:    end for
15: end while
```

Although the identifiers ($v.id$) of a node's replicas across



(a) $\mathcal{W}$ (b) $G(V,E)$ (c) $\mathcal{G}(\mathcal{V},\mathcal{E})$ (d) M_c^0

Fig. 2. (a) A sample environment with obstacles decomposed into a grid; (b) Connected communication graph G with the weights in f_C; (c) Directed layered graph $\mathcal{G}$ generated from G and (d) Communication map M_c^0 as a form of a shortest path tree excluding irrelevant nodes of $\mathcal{G}$.

all layers are identical, their cost attributes $v.cost$ differ at different layers. Initially, the cost of the root node $v_s.cost = 0$ and the parents of all the nodes are set to $NULL$ (line 2), as many nodes have multiple incoming edges. Our target is to select one incoming edge per node in order to choose a parent. We dequeue a node u from Q (line 6) and check to see if setting it as the parent of its neighbors in the next layer will reduce their costs. Accordingly, in lines 7-14 we select node $u \in \mathcal{V}$ as the parent of a node $v \in \mathcal{V}$ if the condition $u.cost + f_C(u,v) < h[v.id]$ is satisfied. Otherwise, $u.cost + f_C(u,v) \geq h[v.id]$ indicates that we already have achieved the optimal cost in one of the prior layers, including the current layer, with a better parent than u. For example, in Figure 2(d) node 2 achieves the optimal cost $h[v_2.id] = 2$ (using (4)) at layer l_2 through the node 1 of layer l_1. During the evaluation of node 2's replica in layer l_3, we do not find any node u that satisfies $u.cost + f_C(u,v_2) < h[v_2.id]$, thus it is excluded from the tree, having no incoming edge. Finally, we achieve a communication map M_c^s, as shown in Figure 2(d), after traversing all the nodes.

Chain Extraction: Given the position of a mobile unit r, and a number of relay robots m, we search for $v \in \mathcal{V}$ s.t. $v.x = r.x$ and $v.y = r.y$ in the $(m+1)$-th layer of the communication map M_c^s. If such a node is found we backtrack recursively using its parent pointer until the root v_s is reached at layer l_0. The nodes found along this traversal are the positions of the intermediate relays. However, if r is not found in layer $(m+1)$, we search for it in layer m, then layer $(m-1)$ and so on, until we find r or reach layer l_1. If r is found in a lower layer $l_{m'}$ where $m' \leq m$, we can achieve a minimum cost using $m'-1$ relay nodes. On the other hand, if we reach layer l_1 in this process, then the position r cannot be served by m relay robots with the current grid resolution and a failure is reported. In this way, our solution is resolution complete [14]. The grid points that cannot be served by m relays compose the *shadow region* $\Phi^m \subset \Omega$ of $\mathcal{W}$:

$$\Phi^m = \{g \in \Omega | g \equiv v \notin l_i \text{ where } 1 \leq i \leq m+1\} \quad (8)$$

In Figure 2(a), grid point 5 cannot be served by $m = 1$ relay and therefore does not appear in the layers l_1 and l_2 of M_c^s in Figure 2(d).

Algorithm analysis: The running time of Algorithm 1 is $O(\mathcal{V}+\mathcal{E})$ as every node and edge is visited once [5]. However, the input is a graph G of n nodes from which we computed $\mathcal{G}$ with $(m+1)(n-1)+1$ nodes for $m+2$ layers. In the worst case, where every node can communicate to all other nodes, the total number of edges is at most $|\mathcal{E}| =$ (number of edges in $m+1$ layers) + (number of edges in layer l_0) = $m(n-1)(n-2)+n-1 = O(mn^2)$, which is also the running time of Algorithm 1.

B. Multiple Unit Multiple Relay Placement

According to the definition of the MULTI-RELAY MULTI-UNIT problem (Problem 2), there are m relays available for serving p mobile units that are located at $r_1, r_2, \ldots, r_p$. We need to compute the optimal locations $q_1, q_2, \ldots, q_m$ that will connect the operator position s to the units. However, the general problem on a plane becomes NP-Hard.

Proposition 4.2: The MULTI-RELAY MULTI-UNIT SERVING problem in a polygon with holes is NP-Hard.

Proof: (Sketch) A Euclidean m-median problem is to find a set of m points on a plane to serve p fixed nodes so as to minimize $\sum_{1\leq i\leq p} \min_{1\leq j\leq m} d(r_i, q_j)$. This is shown as NP-Hard in [20] and [21] for polygons with holes. Our MULTI-RELAY MULTI-UNIT problem is similar except that the $m+p+1$ points need to form a connected component, and therefore cannot be relaxed to an easier version. Thus, according to the technique of proof by restriction [22], MULTI-RELAY MULTI-UNIT contains the Euclidean m-median problem and is therefore NP-Hard. ■

Consequently, we use the same discretization method of relay chain placement similar to that shown in Figure 2(a). We need to compute a minimum spanning sub-tree of $G(V,E)$ (Figure 2(b)) that spans over all the p unit locations, m relays, and the operator such that the units become the leaf nodes while all the relays become the internal nodes. The problem of interest has commonalities to the limited branching *Steiner* tree discussed in [3] where the authors prove that a polynomial time algorithm exists for a fixed number of branching and terminal nodes (m intermediate and p terminals in our case). However, we must prevent the remote units from branching and must make the operator the root.

Algorithm 2 computes the solution for the optimal (for a given resolution) multi-relay positioning for multiple units. Let the set of $p+1$ fixed nodes be $V_T = \{v_s\} \cup V_B$, where $v_s \in V$ is the operator node and $V_B \subset V$ is the set of nodes corresponding to the remote units. As we have n nodes in

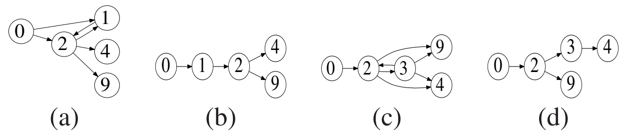


Fig. 3. The operator is at cell 0 and two units ($p = 2$) are placed at cells 4 and 9 that need to be served by $m = 2$ available relays: (a) A sub-graph G_1 constructed with $\nu_1 = \{v_1, v_2\}$; (b) Resulting min-arborescence tree T_1 of G_1; (c) Another candidate sub-graph G_2 with $\nu_1 = \{v_2, v_3\}$; and (d) Candidate tree T_2

$G(V, E)$ (from n grid points), including the $p+1$ fixed nodes, we have to select m relay locations from the remaining $n - p - 1$ nodes. Therefore, we define $\vartheta_m \subset \mathcal{P}(V \setminus V_T)$ as the set of all possible sets of nodes with exactly m members. Here, $\mathcal{P}(V \setminus V_T)$ is the power set of the remaining nodes other than the fixed nodes. Accordingly, ϑ_m has $\binom{n-p-1}{m}$ members that are used to enumerate $\binom{n-p-1}{m}$ possible graphs, each of which has exactly m relays, p units, and one operator.

Algorithm 2 multiRelayMultiUnit($G(V, E)$)

```
1:  V_T = {v_s} ∪ V_B
2:  ϑ_m = {ν ∈ P(V \ V_T) : |ϑ| = m}
3:  for ν_i ∈ ϑ_m do
4:      V_i = ν_i ∪ V_T
5:      G_i = computeDiGraph(V_i)
6:      if G_i.connected() then
7:          T_i = minArborescence(G_i)
8:          T.add(T_i)
9:      end if
10: end for
11: return failure if T = Null
12: return argmin_{T_i ∈ T} [f_C^T(T_i)]
```

From lines 3-10 of Algorithm 2, we compute a set of $\binom{n-p-1}{m}$ spanning trees, $\mathcal{T}$, and select the optimal one. For each set $\nu_i \in \vartheta_m$ of m nodes, we construct a directed sub-graph $G_i(V_i, E_i)$ from the undirected graph $G(V, E)$, where $V_i \subset V$ and $V_i = \nu_i \cup V_T$ (in total, $m+p+1$ nodes). For each undirected edge $(u, v) \in E$, if $u, v \notin V_B$, then the edge is replaced with two directed edges. Otherwise, if $u \in V_B$, then the edge is replaced with only one directed edge from v to u, or vice versa (see Figure 3(a) and (c)). Also, the operator node (root) v_s has no incoming edges.

$$E_i = \{(u, v) \in E : u \notin V_B \text{ and } v \neq v_s \text{ where } u, v \in V_i\} \tag{9}$$

Once we construct a graph G_i, we check its connectivity and exclude it from further computation if it is not connected. Otherwise, on the graph G_i that has exactly m relays, p units, and one operator, we compute the minimum spanning tree T_i which is generally called the min-*arborescence* tree [4] for directed graphs. We apply Tarjan's algorithm [4] to get a minimum arborescence tree T_i (see Figures 3(b) and (d)). Finally, we choose the tree that yields the minimum cost: $\underset{T_i \in \mathcal{T}}{\operatorname{argmin}} f_C^T(T_i)$, in line 12 of Algorithm 2.

Algorithm analysis: The running time of Algorithm 2 depends on lines 3-10. The *loop* of line 3 runs $\binom{n-p-1}{m}$ times, which can be simplified as $\frac{(n-p-1)^m}{m!} = \frac{(n-p-1)(n-p-2)\ldots(n-p-1-m)}{m(m-1)\ldots 2\cdot 1} = O(n^m)$ for a constant m. As the Tarjan's algorithms runs in $O(E + V \log V)$ [4], in the worst case it's running time is $O(E) = O(m + p + m(m-1) + mp) = O(m^2 + mp + p)$ (by sub-graph construction as shown in Figures 3(a) and (c)). Therefore, the running time of Algorithm 2 is $O(n^m(m^2+mp+p))$. Generally, for a robotic mission, the given number of relays, m is fixed which makes the running time polynomial.

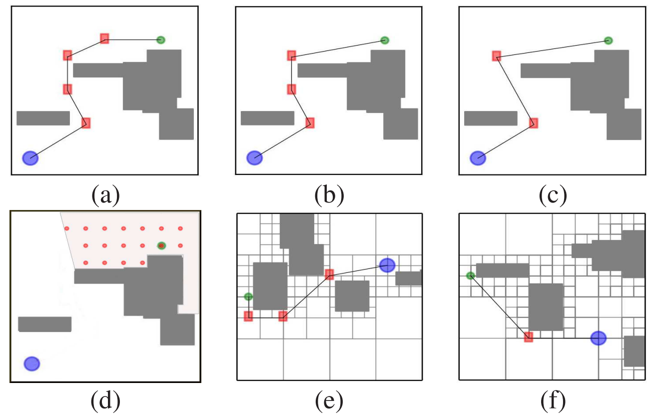


Fig. 4. Multi relay chain simulation: (a) Four relays forming a chain; (b) and (c) Number of relays are reduced to three and two, respectively; (d) Shadow region Φ^1 for one relay (using (8)); (e) and (f) Adaptive grid decomposition for three relays and one relay.

V. Experimental Results

A. Software Simulation

Multi-Relay Chain: We have implemented Algorithm 1 on several randomly generated environments shown in Figures 4 and 5. Figure 4(a) is a solution to a visibility based system when we have 4 intermediate relays and no connection is allowed through the obstacles $\mathcal{O}$. In this case, we search and find the unit node in layer l_5 of the communication map M_c^s and backtrack until we reach the operator node in layer l_0. Next, we reduce the number of relays to 3 and then 2, and the solutions extracted from the same M_c^s are shown in Figure 4(b) and (c), respectively. In all of the cases, our algorithm extracted solutions form the same map M_c^s and are able to minimize the distances of the successive nodes in the chains.

Figure 4(d) shows a case where a single relay cannot serve the unit which stays inside the shadow region Φ^1 (as per (8)). In Figures 4(e) and (f), we demonstrate scenarios where quad-tree based adaptive grids [23] are used instead of uniform grid. This type of grid generates more grid points around the obstacles and less points in the obstacle free space and therefore may reduce the number of computation by Algorithm 1. However, the quality of solution may not be as good as the uniform sampling, specially where the free space is large.

In our next case study, we allow the signal to be penetrated through the obstacles $\mathcal{O}$ (like radio waves). The path loss is therefore impacted by fading and diffraction effects that are modeled by the obstacle crossings function, $f_{\mathcal{O}}$ (as per (2)). We have measured the average value of $f_{\mathcal{O}} = 9$ dB through trial experiments for single wall crossings using the XBee communication device. Accordingly, Figures 5(a), (b) and (c) show the optimal relay placements for four, three, and one available relays, respectively. In all three cases, the algorithm

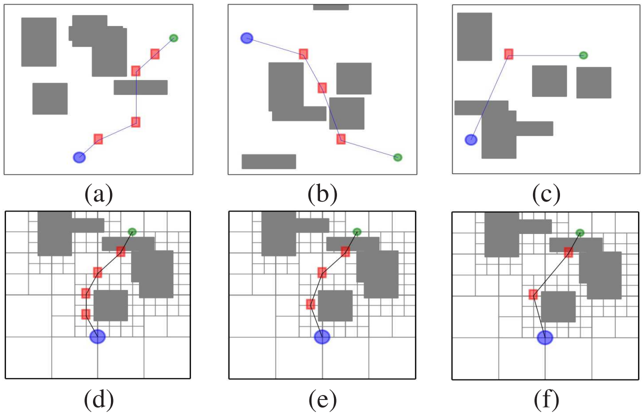


Fig. 5. Communication can now be established through the obstacles with extra costs according to (2): (a) Four relays, (b) three relays and (c) one relay connecting the unit to the operator. (d), (e) and (f) are the adaptive grid decomposition with four, three and two available relays, respectively.

found the minimal obstacle intersections to minimize the communication cost f_C^L. In Figures 5(d), (e) and (f), the adaptive grid decomposition are applied, and we compute the solutions for four, three, and two available relays. We see that the average link lengths are increasing with the decreasing number of relays, which results in reduction of signal strength.

Multi-Relay Multi-Unit Tree: Six sample min-arborescence trees are shown in Figure 6 as generated by Algorithm 2 with both the uniform and adaptive grid discretizations. The operator does not directly serve the units, which means the units receive their service from one of the relays. Figures 6(a) and (b) demonstrate the cases of two vehicles A_1, and A_2 relaying communication to four and six units, accordingly. Next, we increase the number of relays to three and four; the outputs are shown in Figures 6(c), (d) and Figures 6(e), (f), respectively. We also observe that the degrees of the relays are distributed evenly in almost all the scenarios that helps to control the overflow of packets through a single relay.

B. Hardware Experiment

To demonstrate our chained-relay, and multi-relay multi-unit solutions, we use robots based on the open source SERB robot [24] in a hardware/software test-bed.

Motion Planning: We used the A^* search algorithm [14] to generate trajectories for relocation of the relay robots. However, the relocation cost (e.g. fuel consumption, navigation costs for large military vehicles) may outweigh the new communication cost, if the communication quality improvement is marginal and the system is currently connected. Therefore, the operator will decide either to move the relays or to stay with the current setup considering the different costs.

MULTI-RELAY CHAIN: In Figure 7 we demonstrate the relay chain and tree formations using two available relay robots A_1 and A_2. A single unit is located at the top-left corner while the operator S stays near the red obstacle as shown in Figure 7(a). In Figure 7(a), robot A_1 starts moving following the path generated by the A^* algorithm, avoiding all the obstacles and other robots. Similarly, A_2 completes its path and reaches its goal location as shown in Figure 7(b), establishing a relay chain (yellow dotted lines).

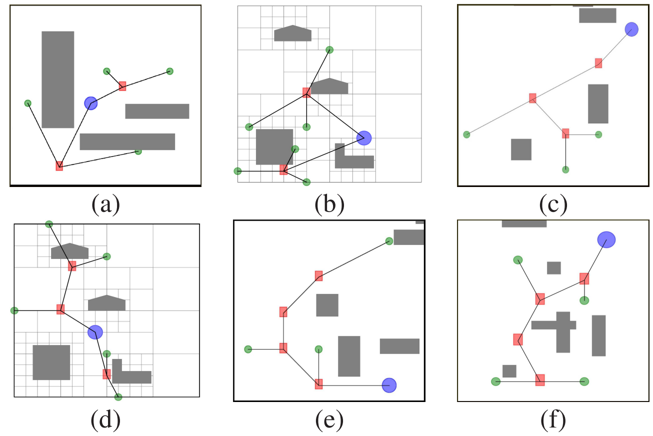


Fig. 6. Multi-Relay Multi-Unit simulations. (a) and (b) show min-arborescence tree for two relays serving four and six units, respectively; (c) shows three relays serving three units and (d) is a case of three relays connecting five units; (e) and (f) are min-arborescence tree for four relays connecting the units to the operator.

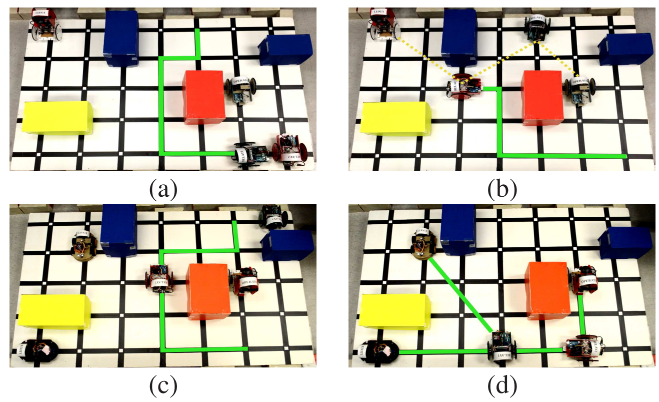
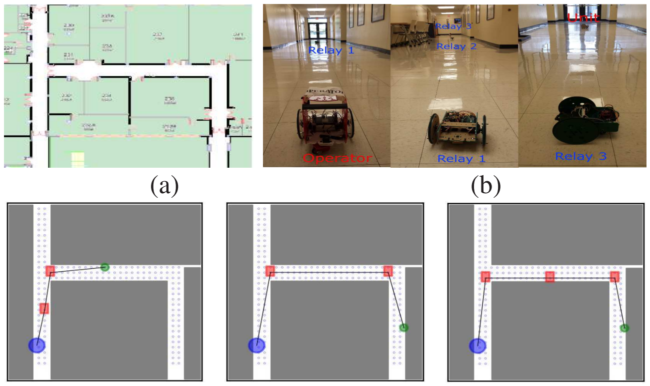


Fig. 7. (a) Multi-relay chain experiment: A_1 is traveling along it's path generated by the A^* algorithm; (b) A_2 reaches its destination and a relay chain is established: $S \rightarrow A_1 \rightarrow A_2 \rightarrow B$; (c) Multi-unit multi-relay experiment: A_2 is moving along its path as generated by the A^* algorithm; (d) A_1 and A_2 reach their destinations and a min-arborescence tree has been formed with the edges $E = \{(v_s, v_2), (v_2, v_1), (v_1, r_1), (v_1, r2)\}$.

MULTI-RELAY MULTI-UNIT: An example environment with two remote units (placed at the top and bottom left) is shown in Figure 7(c) where the vehicle A_2 is moving along its path towards its destination. Finally, in 7(d), both vehicles A_1 and A_2 have reached their destinations and an optimal communication tree (green lines) is established.

LARGE-DEPLOYMENT: We also have conducted experiments inside a corridor of a building as shown in Figure 8. The locations of the two relay robots are computed by our system as shown in Figure 8 (c) and the maximum signal loss has been measured at 23 dB (decibel). Thereafter, the unit moves to a new location and gets disconnected. As a result, we extract the new positions for the relay robots from our same reusable communication map and the relays are relocated accordingly (Figure 8 (d)). The maximum communication cost was measured at 37 dB with this setup. Finally, three relays are deployed and the maximum link cost has been reduced to 25 dB from 37 dB (Figure 8 (e)).



(a) (b)

(c) (d) (e)

Fig. 8. Large area deployment: (a) A corridor map; (b) Placements of relays as a chain formation; (c) Two relays are deployed; (d) The relays are relocated as the unit moved to a new location; (e) Number of relays is increased to three.

TABLE I
ANALYSIS OF RUNNING TIME (IN SECONDS)

Nodes	Our Method		Burdakov et el.	
	Building $\mathcal{G}$+ M_c^s computation	Subsequent Runs	Building G + k-hop BF	Subsequent Runs
361	6.70+0.438	0.0052	2.39+1.23	1.05
400	8.15+0.58	0.0067	2.66+1.62	1.50
625	23.82+2.41	0.0081	6.54+4.12	4.06
729	35.57+3.70	0.0079	11.39+7.78	7.23
900	49.86+8.01	0.0095	13.52+8.93	8.85
1089	85.01+14.07	0.012	23.03+14.51	14.87

C. Numerical Results

In Table I, we compare our relay chain model (Algorithm 1) with a closely related solution from [2] in terms of the increasing number of nodes. For each model, the left column has two components: 1) the time to build the graph + 2) the time to compute the underlying data structure (M_c^s in our case), and the right column shows the time to recompute solutions in response to the changes either in the number of relays or the location of the unit. Although our graph-building phase takes longer than that of [2] due to the construction of the layered graph $\mathcal{G}$, computation of reusable map M_c^s is commonly faster for smaller environments as we use a modified BFS algorithm on $\mathcal{G}$ (which is a tree), compared to a modified Bellman-Ford algorithm [5] used on G according to [2]. Then, we achieve significant improvements in the subsequent computations than [2], as we only need to extract a chain of relays from M_c^s instead of recomputing the entire data structure.

VI. CONCLUSION AND FUTURE WORK

We have studied the complexity of optimal relay placement problems in an environment filled with obstacles, and proposed solutions that are capable of dealing with most variations of the problems. In the case where we have multiple relay robots, we build a static map which is a reusable data structure computed from a layered graph using the modified breadth-first search algorithm. Thus, a chain formation can be obtained for m available relays and a single unit in different positions. This eliminates a significant amount of re-computation in scenarios where the unit relocates, the number of relay changes in the same environment. We also developed a solution for optimal placement of multiple relays in order to serve multiple units. The solution is a tree and we generate a number of alternate min-*arborescence* trees from which we select the optimal one in terms of communication cost.

One immediate extension of our work is to test the solutions for different communication modalities and perform a benchmark analysis. We are aware of the running time of the multi-unit problem where all possible combinations of candidate nodes may take a long time in the case of many relays. However, this can be improved by early decomposition of the environment and weeding out the unnecessary nodes.

VII. ACKNOWLEDGMENT

This work was supported in part by ARO grant 67736CSII.

REFERENCES

[1] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnström, and P. Olsson. Positioning unmanned aerial vehicles as communication relays for surveillance tasks. In *Robotics: Science and Systems*, 2009.
[2] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnström, and P. Olsson. Relay positioning for unmanned aerial vehicle surveillance. *The international journal of robotics research*, 29(8):1069–1087, 2010.
[3] D. Watel, M. Weisser, C. Bentz, and D. Barth. Steiner problems with limited number of branching nodes. In *International Colloquium on Structural Information and Communication Complexity*, pages 310–321, 2013.
[4] H. N Gabow, Z. Galil, T. Spencer, and R. E. Tarjan. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, 6(2):109–122, 1986.
[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.
[6] P. Kling and F. Meyer auf der Heide. Convergence of local communication chain strategies via linear transformations: or how to trade locality for speed. In *Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures*, pages 159–166, 2011.
[7] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Elsvier Ad Hoc Networks*, 6(4):329–337, 2001.
[8] D. Kirkpatrick. Optimal search in planar subdivisions. *SIAM Journal on Computing*, 12(1):28–35, 1983.
[9] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, 1987.
[10] J. O'Rourke. Visibility. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry, 2nd Ed.*, pages 643–663. Chapman and Hall/CRC Press, New York, 2004.
[11] W. Ren and N. Sorensen. Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, 56(4):324–333, 2008.
[12] P. Renaud, E. Cervera, and P. Martiner. Towards a reliable vision-based mobile robot formation control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.
[13] X. Li and J. Xiao. Robot formation control in leader-follower motion using direct lyapunov method. *International Journal of Intelligent Control and Systems*, 10(3):244–250, 2005.
[14] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.
[15] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej. Using RSSI value for distance estimation in wireless sensor networks based on zigbee. In *15th IEEE International Conference on Systems, Signals and Image Processing*, 2008.
[16] K. Pahlavan and A. H. Levesque. *Wireless information networks*, volume 93. John Wiley & Sons, 2005.
[17] E. M. Arkin, J. S. B. Mitchell, and C. D. Piatko. Bicriteria shortest path problems in the plane. In *Proceedings of 3rd Canadian Conference on Computational Geometry*, 1991.
[18] J. S.B. Mitchell, C. Piatko, and E. M. Arkin. Computing a shortest k-link path in a polygon. In *Proceedings of the thirty-third annual ACM symposium on Foundations of Computer Science*, 1992.
[19] A. Yershova and S. M. LaValle. Deterministic sampling methods for spheres and SO(3). In *Proceedings IEEE International Conference on Robotics and Automation*, 2004.
[20] N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196, 1984.
[21] S. P. Fekete, J. S.B. Mitchell, and K. Weinbrecht. On the continuous weber and k-median problems. In *Proceedings of the sixteenth annual ACM symposium on Computational geometry*, 2000.
[22] M. R. Gary and D. S. Johnson. Computers and intractability: A guide to the theory of np-completeness, 1979.
[23] F. Pascal and J. L. Marechal. Fast adaptive quadtree mesh generation. In *1998 International Meshing Roundtable*. Citeseer.
[24] L. Bobadilla, F. Martinez, E. Gobst, K. Gossman, and S. M. LaValle. Controlling wild mobile robots using virtual gates and discrete transitions. In *Proceedings IEEE American Control Conference*, 2012.