Decentralized Control for Optimizing Communication with Infeasible Regions

Stephanie Gil, Samuel Prentice, Nicholas Roy and Daniela Rus

Abstract In this paper we present a decentralized gradient-based controller that optimizes communication between mobile aerial vehicles and stationary ground sensor vehicles in an environment with infeasible regions. The formulation of our problem as a MIQP is easily implementable, and we show that the addition of a scaling matrix can improve the range of attainable converged solutions by influencing trajectories to move around infeasible regions. We demonstrate the robustness of the controller in 3D simulation with agent failure, and in 10 trials of a multi-agent hardware experiment with quadrotors and ground sensors in an indoor environment. Lastly, we provide analytical guarantees that our controller strictly minimizes a nonconvex cost along agent trajectories, a desirable property for general multi-agent coordination tasks.

1 Introduction

Decentralized control of robotic systems has enabled complex group behaviors such as rendezvous, formation keeping and coverage to be applied to a wide range of engineering problems; however, the absence of centralized computation increases the demands on communication quality [1–4]. This paper focuses on the problem of optimizing communication quality between a multi-agent network of mobile robots and stationary sensors. In previous work [5] we developed a decentralized gradient-based controller that provably optimizes communication quality amongst the network, but this approach is limited to environments where the entire space is feasible. In practical scenarios, such as the indoor environment shown in Figure 1, there often exist regions of space that are hazardous or untraversable. Such obstacles make designing the controller difficult for two main reasons: 1) the goal state, or optimal communication configuration, is unknown *a priori* and 2) the presence of infeasible regions introduce many constrained local minima that may be less satisfactory solutions. This work uses nonlinear optimization techniques to derive a decentralized controller that is easy to implement and addresses the problem of communication optimization in the presence of infeasible regions.

Massachusetts Institute of Technology, MA. {sgil,prentice,nickroy,rus}@mit.edu

Fig. 1 Multi-agent field test environment with Ascending Technology Pelican quadrotors (solid outlines) and stationary ground sensors (dashed outlines). Infeasible regions include the wall in the center of the room, an open staircase that is partially visible on the right, and a table (out of view).

The introduction of infeasible regions raises many challenges. The cost for the communication optimization problem is nonconvex which is a necessary property of many interesting distributed tasks [6]. Therefore we aim for simple-to-implement controllers that have the desired properties of scalability, reliance only on local information, and that descend the cost along the trajectories of each agent. Gradient-based controllers are thus ideally suited. However, the presence of infeasible regions breaks up the free space into several sets over which we must optimize, introducing challenges both for convergence, and for the quality of the converged solution. As an example, an aerial vehicle may get "stuck" behind a wall that it could easily fly around if the direction of steepest descent of the cost happens to be perpendicular to the obstacle edge as illustrated in Figure 2.b, and so we need to consider a wider range of descent directions to avoid these scenarios. As a result of gradient-based optimization over a nonconvex environment, achievable convergence is either to a critical point of the cost in the best case, or to a point of improved cost on the edge of an infeasible region. Our aim is to derive a controller such that agents descend the cost along the generated trajectories, and where these trajectories are biased towards directions that avoid infeasible regions and thus have a larger range of attainable improved cost solutions.

The use of gradient projection methods from nonlinear optimization [7] allows us to formulate our nonconvex problem as a simple quadratic program where the constraint set is a convex subset of the free space in the environment. We use the solution of a mixed integer program to effectively select the convex feasible region over which we optimize our cost and the result is a mixed integer quadratic program (MIQP) that can be solved efficiently for each agent. We show that the addition of a scaling matrix, that preserves the quadratic and thus efficiently solvable attributes of the problem, allows the designer to influence vehicle trajectories to move around infeasible regions and improve the range of attainable converged solutions. In particular, in Section 3.2.1 we derive analytical results relating the heading angle to the steepest descent direction for a chosen scaling matrix, and we show that we retain descent of the cost. Theorem 3 shows the existence of a sequence of scaling matrices such that our algorithm produces trajectories reaching unconstrained local minima of the communication cost, if such a trajectory exists for the given initial positions and environment. Although the derivation of a sequence of scaling matrices that guarantees convergence to unconstrained local minima of the cost remains an open question, in the Results Section we provide a heuristic selection of scaling matrices that demonstrates good performance in simulation and hardware experiments.

Section 4.1 presents our communication optimization algorithm and demonstrates the performance of the controller and its robustness in the case of agent failure. Lastly, in Section 4.2 we demonstrate our control method on real aerial vehicles that must navigate through an indoor environment (Figure 1) to optimize communication amongst a network of three stationary ground vehicles.

1.1 Related Work

Artificial potential fields for obstacle avoidance as in [8–10], decomposition of the environment using different notions of a graph through which to search the space [11, 12], and shortest path methods as in [13], represent active areas of research for the problem of vehicle coordination in environments with obstacles. In the current work, final positions of the agents are local minima of the communication cost and since this cost must be optimized iteratively, these local minima are *unknown a priori*. Therefore we cannot assume knowledge of final goal states that we can navigate towards, and we cannot disallow minima from being inside of infeasible regions.

The characteristic that the optimization problem itself defines the agent trajectories makes this problem particularly challenging. The paper [14] also addresses a multi-agent optimization problem but for coverage of a 2D environment and uses a clever mapping inversion. Methods similar in spirit to our work are Mixed Integer Methods as in [15, 16], although these methods are different in that they also consider navigation to known goal states. For our work we must also descend the cost along agent trajectories as convergence to local minima of the cost and maintenance of connectivity for the network hinge on this requirement. Thus a strong motivation for this work is to ensure that descent of the cost is achieved at each iteration. The requirement of provably descending the cost along vehicle trajectories is common for many coordination tasks and thus illustrates the generality of the communication optimization problem to other multi-agent tasks [2–4].

2 Problem Formulation

In previous work [5], we derived a cost function that optimizes communication quality among aerial vehicles and ground sensors. This cost uses a Signal to Interference ratio (SIR) to weigh communication strength of a pair of vehicles against interference from neighboring vehicles and is a weighted sum of two terms where the first term maximizes the SIR of each individual link and the second term equalizes SIR over all links. The resulting behavior of the controller is designed by increasing or decreasing ρ, which is a scalar that assigns more or less weight to the second term in the cost.

The cost $H : \mathbb{R}^{(p\times N)} \to \mathbb{R}$ is defined over all vehicle positions ${x^k}_i \in \mathbb{R}^p$ for N vehicles at iteration k as:

$$H({x^k}_1,\ldots,{x^k}_N) = \sum_{i}^{N}\sum_{j\neq i}^{N} -SIR_{ij} + \frac{\rho}{SIR_{ij}+\delta} \tag{1}$$

where i and j is shorthand for vehicles with positions x_i and x_j respectively, and δ is an arbitrarily small positive number to allow $SIR_{ij} = 0$. The Signal-to-Interference Ratio $(SIR) : \mathbb{R}^p \to \mathbb{R}$ is given by $SIR_{ij} = \frac{f_{ij}}{N_i + \sum_{k \in \mathcal{N}_{i \setminus j}} f_{ik}}$ and the signal strength between two communicating agents i and j is given by $f_{ij} : \mathbb{R}^p \to \mathbb{R}$. The signal strength is given by $f_{ij} = \frac{P_0}{d_{ij}^{\beta}}$. All vehicles in the neighborhood of i not including j is denoted $\mathcal{N}_{i \setminus j}$, $P_0 \in \mathbb{R}$ is a given maximum signal strength, β is a given dropoff parameter and often $\beta = 2$, and $d_{ij} = \left\| x_i - x_j \right\|$.

2.1 Communication Optimization as a MIQP

We wish to move N agents along trajectories that descend the cost $H(\mathbf{x}^k)$ from (1), where $\mathbf{x}^k \in \mathbb{R}^{p\text{x}N}$ is the vector of all vehicle positions at time k, while constraining this trajectory to remain outside of infeasible regions for all time. For each vehicle i with position $x_i^k \in \mathbb{R}^p$ at iteration k, we wish to move an amount $s^k > 0$ along the direction of steepest descent,$-\nabla H(x^k)$, but we must enforce the constraint to stay within free space. The value $\nabla_i H(x^k) \in \mathbb{R}^p$ is the gradient of the cost H with respect to the position of vehicle x_i at time k and we note that although the cost is global, the derivative $\nabla_i H(x^k)$ depends only on *local* information and is distributed in this sense. We subsequently drop the subscript i to simplify notation so that x^k is the position of vehicle i and $\nabla H(x^k)$ refers to the gradient of H with respect to the position of agent i.

Gradient projection methods from nonlinear optimization allow us to formulate descent for our nonconvex cost while maintaining the constraint of staying a convex set. Because the free space of the environment is almost never convex we must divide the free space set into the intersection of many convex sets, which is possible in particular for environments with convex polygonal infeasible regions which is the case that we consider. We take advantage of the fact that each agent needs to optimize H only over its local environment and employ a mixed integer program to activate a local convex subset of the free space over which we can perform gradient projection. The result is a Mixed Integer Quadratic Program (MIQP):

$$\begin{aligned} \min_{x,t} \; & \left\| x - ({x_i}^k - s^k \nabla_i H(x^k)) \right\| \qquad (2) \\ s.t. \; & A_l x \leq b_l + t_l M, \;\; \forall l \in \{1, \ldots, L\} \\ & \sum_{j=1}^{E_l} t_{lj} \leq E_l - 1, \; \forall l \in \{1, \ldots, L\} \\ & t_{lj} \in \{0, 1\} \;\; \forall j, l \end{aligned}$$

where s^k is a scalar > 0, L is the number of polygonal infeasible regions in the environment, E_l is the number of edges for infeasible region l, M is a sufficiently large scalar, and $t_l \in \mathbb{R}^{E_l}$ is a binary column vector returned by the MIQP for each infeasible region, and $A_l \in \mathbb{R}^{(E_l \times p)}, b_l \in \mathbb{R}^{E_l}$ describe the convex,polygonal, infeasible regions as defined next. We now provide the mathematical descriptions of infeasible regions and free space sets returned as solutions from the MIQP:

Definition 1 ***Infeasible Regions and Free Space Sets:*** Infeasible regions *are convex, polygonal sets that are the intersection of E_l halfspaces $\bigcap_{i=1}^{E_l}(A_i x \geq b_i)$. A vehicle may not move through an infeasible region but we assume communication strength is not affected. The binary column vectors from* (2) *encode feasible region constraints and thus a particular solution of binary variables $t^* \in \mathbb{R}^{LE_l}$ effectively "activates" one or more edges of each infeasible region such that these selected edges are the valid constraints enforced in solving the MIQP. The intersection of the halfspaces corresponding to the activated obstacle edges is always a closed and convex set denoted*

$$X_{F_{t^*}} = X_F(t^*) = \{x | A_l x \leq b_l + t_l^* M\} \tag{3}$$

where $X_F(t^*)$ is the closed, convex free space subset *corresponding to the binary variable solution t^* of Equation* (2)*.*

The intuition for the formulation in Equation (2) is that each vehicle moves as far along the direction of steepest descent of the cost H as possible while staying within feasible space. The problem with this formulation however, is that if the direction of steepest descent becomes perpendicular to the edge of an infeasible region then it is possible to get stuck behind this edge even in the case where the vehicle may be able to easily go around the obstacle by moving along a descent direction that is not that of steepest descent, see Figure 2.b. We address this problem in the formulation of the next section.

2.2 Use of Scaling to Avoid Regions of Infeasibility

The MIQP formulation from the last section can be solved efficiently using off-the-shelf optimizers, and results in a very simple form of a controller but suffers from the limitation of always following the steepest descent direction, even in the case where this direction is obstructed by an infeasible region. Thus, we wish to improve the range of attainable solutions while conserving the simplicity of the MIQP from the previous section. To this aim we propose use of the *scaled* gradient projection method.

In nonlinear optimization theory the scaled gradient projection method is often used to improve rate of convergence [7]. Our objective, however, is to influence the vehicle trajectory towards directions that are not perpendicular to active constraint edges. In addition, the scaled gradient projection method amounts to the addition of a term that is quadratic in the optimization variable x and thus is also a quadratic program as in the previous case and can be easily solved. We define a new problem whose optimization results in a feasible waypoint $\bar{x}_S^k$ for agent i (where i subscripts are dropped):

$$\begin{aligned} \bar{x}_S^k = \arg\min_x & \left\| x - z^k \right\|_{S^k} \\ \text{s.t. } & A_l x \leq b_l + t_l M \quad \forall l \\ & \sum_{j=1}^{E_l} t_{lj} \leq E_l - 1 \quad \forall l \in 1, \ldots, L \\ & t_{lj} \in \{0, 1\} \quad \forall j, l \end{aligned} \tag{4}$$

where the matrix $S^k \in \mathbb{R}^{p\times p}$ is a positive definite matrix, and we use the notation $\|q\|_{S^k} = q'S^k q \ \forall q \in \mathbb{R}^p$ to represent the scaled norm. For this scaled formulation, the desired waypoint is

$$z^k = x^k - s^k (S^k)^{-1} \nabla H(x^k) \tag{5}$$

A more compact definition of (4) can be written using the representation of the free space set from (3) for each vector of binary variables t^* that solve (4):

$$\vec{x}_S^k = \arg \min_{x \in X_F(t^*)} \left\| z^k - x \right\|_{S^k} \tag{6}$$

The position update rule for x^{k+1} is given by:

$$x^{k+1} = x^k + \alpha^k (\vec{x}_S^k - x^k) = x^k + \alpha^k d^k \tag{7}$$

Where the stepsizes α^k and s^k satisfy Assumption 1:

Assumption 1 *There exist stepsizes $\alpha^k > 0$ and $s^k > 0$ that are sufficiently small such that given a descent direction d^k, a step along this direction will not intersect an obstacle and will provide sufficient decrease of the cost in the sense of the Armijo, or limited minimization rule that are standard in Nonlinear Programming [7]. We assume that α^k and s^k satisfy these conditions throughout the paper.*

A first order Taylor series expansion of the cost around the current point x^k shows that descent of the cost is possible for small enough stepsize along a valid descent direction d^k. In the case that the current iterate is at the edge of an obstacle, the stepsize would necessarily be zero to avoid intersecting the obstacle and the method will stop. The requirement that S^k is positive definite is necessary to maintain descent of the cost $H(\mathbf{x}^{k+1}) < H(\mathbf{x}^k)$. In effect, our next waypoint x^{k+1} will minimize distance in the sense of the scaled norm to our desired waypoint z^k [7]. See Figure 2.

We define the descent direction $d^k = \vec{x}_S^k - x^k$. The advantage is that now we can steer our trajectory to any heading relative to the direction of steepest descent $-\nabla H(x^k)$, as long as this direction satisfies $d^k = \{(\vec{x}_S^k - x^k) | \ (\vec{x}_S^k - x^k)' \nabla H(x^k) < 0\}$ where $x'y$ is the dot product of a vector x and a vector y, and the achieved relative heading angle θ depicted in Figure 2 is defined as:

$$\theta = \arccos \left\{ \frac{(-\nabla H(x^k))' d^k}{\|\nabla H(x^k)\| \, \|d^k\|} \right\} \tag{8}$$

We use this flexibility to assign preference to paths that entirely clear regions of infeasibility that are in the direction of the negative gradient. In Section 3 we derive an analytical relationship between S^k and θ.

3 Analysis

3.1 Analysis of the Unscaled Controller

We show that the sequence of vehicle positions produced by the MIQP, in combination with the update rule from (7) produces strict descent directions such that $H(\mathbf{x}^{\mathbf{k+1}}) < H(\mathbf{x}^{\mathbf{k}})$ for all k for stepsizes satisfying Assumption 1. Proving descent of the cost, such that $H(\mathbf{x}^{k+1}) - H(\mathbf{x}^k) < 0$, is made challenging by the general non-uniqueness of the solution for the binary variables t in Equation (2). This in turn means that the convex subset over which we perform optimization may not be unique and may not contain the current iterate x^k which makes the classical descent proof for gradient projection methods not applicable.

From the result asserting that the cost is reduced at each iteration, and the fact that the local minima of H are finite as shown in previous work, [5], we expect convergence to a fixed point. This fixed point can either be at the edge of an infeasible region where the projection $\vec{x}^k = x^k$ (stationary) as defined in Lemma 1.4 or can be a critical point of the cost H. In the following section we show how scaling can be used to decrease the likelihood of getting "stuck" at the side of an infeasible region.

We use the concept of a vector d being *gradient-related*.

Definition 2 ***Gradient Related:*** *A bounded direction sequence $\{d^k\}_{k\in\mathscr{K}}$ is gradient-related to any subsequence $\{x^k\}_{k\in\mathscr{K}}$ that converges to a nonstationary point, if:*

$$\lim_{k\to\infty} \sup_{k\in\kappa} \nabla H(x^k)' d(x^k) < 0 \;\; \forall k \in \mathscr{K} .$$

We use the following properties of projection from [7]:

Lemma 1 (Properties of the projection onto a convex set X) *Let X be nonempty, closed and convex, and let $[z]^+$ denote the projection of $z \in \mathbb{R}^p$ onto X:*

1. *The projection of $z \in \mathbb{R}^p$ exists, is unique, and minimizes $\|z-x\|$ over $x \in X$.*
2. *It must hold that $(z-[z]^+)'(x-[z]^+) \leq 0, \; \forall x \in X$*
3. *The projection function is continuous.*
4. *We have $\tilde{x} = [\tilde{x} - s\nabla H(\tilde{x})]^+$ for all $s > 0$ iff $\tilde{x}$ is stationary.*

We now seek to show that the d^k produced by the solution to (2) are gradient-related for all k and thus for stepsizes satisfying Assumption 1 we have $H(\mathbf{x}^{k+1}) - H(\mathbf{x}^k) < 0$.

Theorem 1 *For the cost H that is differentiable everywhere, the sequence of directions $\{d^k\}$ produced by solving the MIQP formulation and using the provided update rule from* (7) *are directions of descent of the cost such that they satisfy the gradient related property for points x^k that are not stationary points of the cost.*

Proof. The proof is identical to that of Theorem 2 with the scaling matrix set to the identity $S^k = I$. □

3.2 Analysis of the Scaled Gradient Projection Method for Avoidance of Infeasible Regions

We provide analytical results for three main problems related to the scaled version of the MIQP (4). First, we relate the scaling matrix S^k to the relative heading angle θ where this direction is relative to the direction of steepest descent. Second, we show that the use of a scaling matrix generates trajectories for each agent over which the cost is descended at each iteration. Lastly, we show that there exists a sequence $\{S^k\}$ of scaling matrices such that our formulation from (4) generates a trajectory converging to the more desirable *unconstrained* local minima of H if such a trajectory exists for the environment. Although the problem of deriving such a sequence of scaling matrices remains open, in the Results section we provide a heuristic method of generation of scaling matrices that circumvent regions of infeasibility but do not guarantee convergence to unconstrained local minima of H.

3.2.1 Controlling the Relative Heading Angle to Avoid Regions of Infeasibility

In this section we gain insight on how to design the scaling matrix to achieve the desired relative heading angle, θ. From our discussion in 2.2, we require that the scaling matrix S^k must be a positive definite matrix and with an orthonormal choice of eigenvectors $v_i \in \mathbb{R}^p$ we can write S^k in a decomposed form, $S^k = V\Lambda V^T$. $V = [v_1, \ldots, v_p]$ is a matrix of eigenvectors of S^k and $\Lambda \in \mathbb{R}^{p\times p}$ is a diagonal matrix of eigenvalues $\{\lambda_1, \ldots, \lambda_p\}$ of S^k. Furthermore we know that that all $\lambda_i > 0$ since S^k is positive definite. Therefore we can write any vector $\in \mathbb{R}^p$, in particular the negative gradient vector $-\nabla H(x_k)$, as a linear combination of the v_i's. In particular, we can write $-\nabla H(x_k) = \sum_{i=1}^p \zeta_i v_i$ where ζ_i are scalars representing the component of $-\nabla H(x^k)$ in the direction of v_i, and we consider normalized eigenvectors such that $\|v_i\| = 1$. By the Pythagorean theorem, and the fact that the v_i are orthogonal, we have that

$$\left\|\nabla H(x^k)\right\|^2 = \sum_{i=1}^{p}(\zeta_i v_i)^T(\zeta_i v_i) = \sum_{i=1}^{p}(\zeta_i)^2 \tag{9}$$

We denote the unprojected heading direction $\tilde{d}^k$, and note that this is $\tilde{d}^k = (z^k - x^k)$ for the scaled gradient projection. From (5) we see that this is simply $s^k(S^k)^{-1}\nabla H(x_k)$. If we again use Pythagorean Theorem to write the expression for $\left\|\tilde{d}^k\right\|$, and the dot product $\nabla H(x^k)'\tilde{d}^k$ we get:

$$\left\|\tilde{d}^k\right\|^2 = \left\|s^k \sum_{i=1}^{p}\frac{1}{\lambda_i}\zeta_i v_i\right\|^2 = (s_k)^2\sum_{i=1}^{p}(\frac{1}{\lambda_i})^2\zeta_i^2 \tag{10}$$

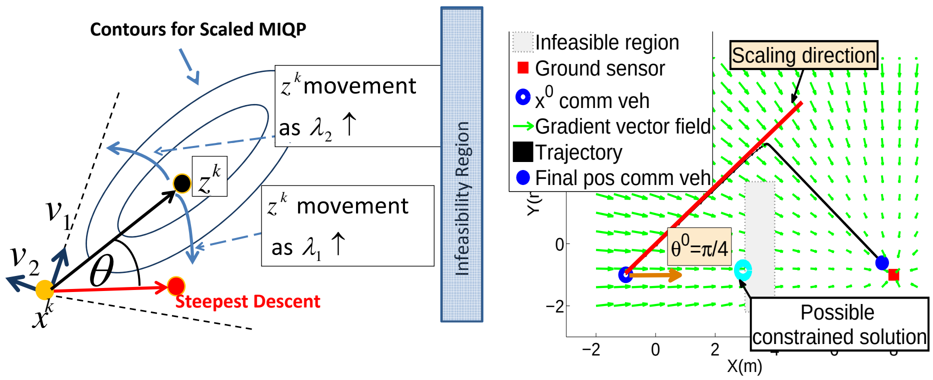
$$\nabla H(x^k)'\tilde{d}^k = s_k\sum_{i=1}^{p}(\frac{1}{\lambda_i})\zeta_i^2 \tag{11}$$

Using the definition of the dot product and the definitions (10), (11), and (9), we get an expression relating the relative heading angle θ to the scaling matrix S^k via its eigenvectors and eigenvalues:

$$\cos(\theta) = \frac{(-\nabla H(x_k))'\tilde{d}^k}{\|\nabla H(x_k)\|\,\|\tilde{d}^k\|} = \frac{\sum_{i=1}^{p}(\frac{1}{\lambda_i})\zeta_i^2}{\sqrt{\left(\sum_{i=1}^{p}(\frac{1}{\lambda_i})^2\zeta_i^2\right)\left(\sum_{i=1}^{p}(\zeta_i)^2\right)}} \tag{12}$$

This expression shows that the scaling matrix can be designed to achieve a specific relative heading angle by careful choice of its eigenvectors and eigenvalues. In particular we notice that if $S^k = I$ where I is the identity matrix and $\lambda_i = 1\ \forall i$, then $\cos(\theta) = 1$, the heading angle is zero and we move in the direction of steepest descent as expected. Alternatively, putting a larger weight on the eigenvalues λ_j of S^k such that $\lambda_j >> \lambda_i,\ \forall j \neq i$, will achieve the effect of causing the heading direction

$\tilde{d}^k$ to align itself most with the component of $-\nabla H(x^k)$ along v_i. See Figure 2 for a schematic of a two dimensional case. However, as the ratio of λ_i gets larger, rate of convergence becomes slower so in general the heading angle θ should not be made larger than necessary.



(a) Schematic. (b) Simulation using scaled MIQP (4).

Fig. 2 Schematic showing the scaled direction and heading angle θ and the change of heading direction as the eigenvalues of S^k are changed 3(a). Simulation of basic scenario showing the utility of scaling to avoid getting stuck behind the wall as in 3(b). Here, a constant scaling matrix $S^k = S$ is used. When the scaled direction reaches a perpendicular angle to the gradient,the trajectory moves along the steepest descent direction as discussed in 3.2.1.

Lastly, a result of (12) is that the direction $\tilde{d}^k$ can never be perpendicular to the negative gradient for a positive definite scaling matrix S^k. As the eigenvector v_i approaches the perpendicular direction to $-\nabla H(x^k)$, the component of the negative gradient vector along this direction $\zeta_i \to 0$ and thus, as seen from Equation (10), $\tilde{d}^k$ cannot be made to move in a direction that is perpendicular to $-\nabla H(x^k)$.

3.2.2 Analysis for the Scaled Gradient Projection Method

In the last section we showed that the addition of a scaling matrix S^k allows us the flexibility to design the relative heading angle to avoid regions of infeasibility in the environment. We now show that the resulting directions $d^k = \vec{x}_S^k - x^k$ where $\vec{x}_S^k$ is solved for from Equation (4), are descent directions such that the cost is reduced over agent trajectories. As in the unscaled case, finding descent directions d^k for our problem is made challening due to the general non-uniqueness of the binary t variables in (4). From the definition of gradient relatedness, the desired property we wish to show is that $\nabla H(x^k)'d^k < 0$ for all k and all solutions d^k at iteration k. The intuition for our proof method is to use the solution d_1^k which is defined for the convex subset containing the current iterate and which can be shown to always be gradient related, to bound all other solutions d_t^k that result from different solutions for the binary variables. From here we can show that $\nabla H(x^k)'d_t^k < 0$ for all d_t^k and k.Using the result of d^k being gradient related for all k, combined with Assumption 1, we get descent of the cost at each iteration such that $H(\mathbf{x}^{k+1}) < H(\mathbf{x}^k)$.

To avoid cumbersome notation, we subsequently drop the S subscript from $\overline{x}_S^k$ and the reader should assume all projections $\overline{x}^k$ in this section are scaled projections. We refer to a point x^k as not stationary if it is not equal to its projection such that $\overline{x}^k \neq x^k$.

Theorem 2 *For the cost $H : \mathbb{R}^{(p\times N)} \to \mathbb{R}$ that is differentiable everywhere, denoting the sequence of vehicle positions $\{x^k\}$ produced by solving the scaled MIQP formulation in* (4) *and using the provided update rule from* (7)*, we have that all directions d^k are directions of descent of the cost such that they satisfy the gradient related property for all x^k not stationary.*

Proof. We denote the set containing the current iterate x^k as X_{F_1}, and the projection of z^k onto X_{F_1} as $\overline{x}_1^k$ and note that this is a solution of the scaled MIQP (6) over the set X_{F_1}. From Lemma 1.2 generalized to scaled projections, it holds that $(z^k - \overline{x}_1^k)'S^k(x^k - \overline{x}_1^k) \leq 0$ for $x^k \in X_{F_1}$ and $\overline{x}_1^k \in X_{F_1}$. Expanding out this property and using the definition of z^k, continuity of the projection, and the fact that we are considering projection onto a single set X_{F_1} containing the current iterate x^k,we have that for all k (see [7]):

$$s^k \nabla H(x^k)'(\overline{x}_1^k - x^k) + \left\| x^k - \overline{x}_1^k \right\|_{S^k}^2 \leq 0, \ \forall k \tag{13}$$

The term $\left\|x^k - \overline{x}_1^k\right\|_{S^k}^2 > 0$ for all S^k positive definite and x^k nonstationary such that $x^k \neq \overline{x}_1^k$. Thus from (13), and Lemma 1.4, we have:

$$\left\| x^k - \overline{x}_1^k \right\|_{S^k}^2 > 0 \Rightarrow s^k \nabla H(x^k)'(\overline{x}_1^k - x^k) < 0. \tag{14}$$

So that the direction d_1^k is always gradient related for x^k and $\overline{x}_1^k$ in the same convex subset, where x^k is nonstationary. We will use this inequality again later. We aim to prove that all directions $d_t^k = \overline{x}_t^k - x^k$ produced from the solutions of Equation (4) satisfy:

$$\nabla H(x^k)'(\overline{x}_t^k - x^k) < 0 \ \ \forall k \tag{15}$$

Because $\overline{x}_1^k$ is a valid solution to the projection of z^k onto X_{F1}, we know that any solution, $\overline{x_t}^k$, to the scaled MIQP in (4) must be within the elliptical set defined by $\overline{x}_1^k$:

$$\mathscr{E}_1 = \{c | (c - z^k)'S^k(c - z^k) \leq r_S\}, \ r_S := (\overline{x}_1^k - z^k)S^k(\overline{x}_1^k - z^k) \tag{16}$$

Now we can write the gradient related condition that we wish to prove as:

$$\begin{aligned} -f^* = &\min_{\overline{x_t}} (-\nabla H(x^k))'(\overline{x_t} - x^k) \\ &s.t. \ \overline{x_t} \in \mathscr{E}_1 \end{aligned} \tag{17}$$

Where our desired condition is that $-f^* > 0$ which ensures that the direction $(\overline{x_t} - x^k)$ is gradient related. The minimization problem written above is well-posed in that f is a continuous function minimized over a compact set $\mathscr{E}_1$ and thus there exists a minimum. Furthermore, this problem can be solved in closed form using Lagrange multipliers to yield the condition:

$$-f^* = -r_S + \left\| s^k \nabla H(x^k) \right\|^2_{S^{k-1}} > 0 \tag{18}$$

If we take this a step further and substitute in the definition for r_S from (16), multiply through by (-1), expand, and simplify we get a new form for the inequality condition that we wish to prove:

$$2s^k \nabla H(x^k)'(\bar{x}_1^k - x^k) + (\bar{x}_1^k - x^k)' S^k (\bar{x}_1^k - x^k) < 0 \; \forall k \tag{19}$$

We compare to the condition (14). From the reasoning shown in (14), we know that $2s^k \nabla H(x^k)'(\bar{x}_1^k - x^k) < 0$ for $\bar{x}_1^k \neq x^k$ which is true by the nonstationary assumption. Thus we have that this desired inequality always holds and all produced d^k are descent directions as desired and this completes the proof. □

To gain more intuition notice that the condition in Equation (19) is equivalent to requiring that $r_S = (\bar{x}_1^k - z^k)S^k(\bar{x}_1^k - z^k) < r_{S\text{cr}} = \left\| s^k \nabla H(x^k) \right\|^2_{S^{k-1}}$. Intuitively what this means is that $\bar{x}_1^k$ is a valid projection of the desired waypoint z^k where the distance to z^k is smaller from $\bar{x}_1^k$ than from x^k in the scaled norm sense, such that $(\bar{x}_1^k - z^k)S^k(\bar{x}_1^k - z^k) < (x^k - z^k)S^k(x^k - z^k)$.

Because we attain descent of the cost at each iteration, and we are optimizing a continuous function over a compact set so that minima are well defined as shown in [5],we therefore expect convergence to a fixed point. This point can be at the edge of an infeasible region or at a critical point of the cost, although the use of scaling aims to circumvent those infeasible regions which do not contain local minima in their interiors.

3.3 Existence of Optimal Sequence of Scaling Matrices

Because we optimize a nonconvex cost, we target convergence to local minima. For the case where these local minima are reachable in feasible space, we consider a sequence of scaling matrices $\{S^k\}$ to be "optimal" if the controller resulting from using Algorithm 1 generates trajectories for all vehicles that converge to an unconstrained local minimum of H. The existence problem is to assert that if there exists such a trajectory for the given environment, then there also exists a sequence of scaling matrices such that the trajectory generated by Algorithm 1 is optimal. We do not find such a sequence, this remains an interesting open question. Instead, we prove the positive result for the existence problem.

Theorem 3 *If $\exists\{g^k\} \to x^*{}_{unc}$,where $\{g^k\}$ is a* valid *sequence of waypoints for each vehicle that converges to an unconstrained local minimum,$x^*{}_{unc}$, of H given $\mathbf{x}^0$, then $\exists\{S^k\}\;\; s.t. \{x^k\} \to x^*{}_{unc}$, where $\{x^k\}$ is the trajectory sequence generated by using Algorithm 1 for each vehicle. A sequence $\{g^k\}$ is* valid *if $H(g^{k+1}) - H(g^k) < 0$ for all k, $g^k \in X_F,\; \forall k$ where X_F is the entire feasible region of the environment, and the stepsize between any consecutive points g^k, g^{k+1} satisfies Assumption 1 and physical vehicle limits.*

Proof. From Proposition 2 we must satisfy $-\nabla H(x^k)'d^k > 0$ for all k. From (11) we see that $\lambda_i > 0$ in order to satisfy this condition. We can write $g^{k+1} - g^k = \sum_{i=1}^p a_i v_i$

for some appropriate a_i since $g^{k+1} - g^k \in \mathbb{R}^p$ and the eigenvectors of S^k span $\mathbb{R}^p$. Since by the descent requirement on $\{g^k\}$ we have $-\nabla H(x^k)'(g^{k+1} - g^k) > 0,\ \forall k$, we can choose orthonormal basis vectors v_i of S^k such that $a_i > 0$ and $\zeta_i > 0$ for all i, where ζ_i are from (9) and thus the choice of $\lambda_i = \frac{s^k}{a_i}\zeta_i$ satisfies $\lambda_i > 0,\ \forall i$ and from the definition of d^k from (10) we see that we can always achieve $d^k = g^{k+1} - g^k,\ \forall k$ for this choice of λ. Since S^k is fully determined through its eigenvectors and eigenvalues as $S^k = V\Lambda V'$ and we have shown that there exists a sequence $\{S^k\}$ for which $\{d^k\} = \{g^{k+1} - g^k\}$ and thus the resulting sequence of agent positions $\{x^k\}$ reaches the unconstrained local minimum of H if $\{g^k\}$ reaches the unconstrained local minimum from given initial positions. □

4 Results

4.1 Algorithm and Simulation Example

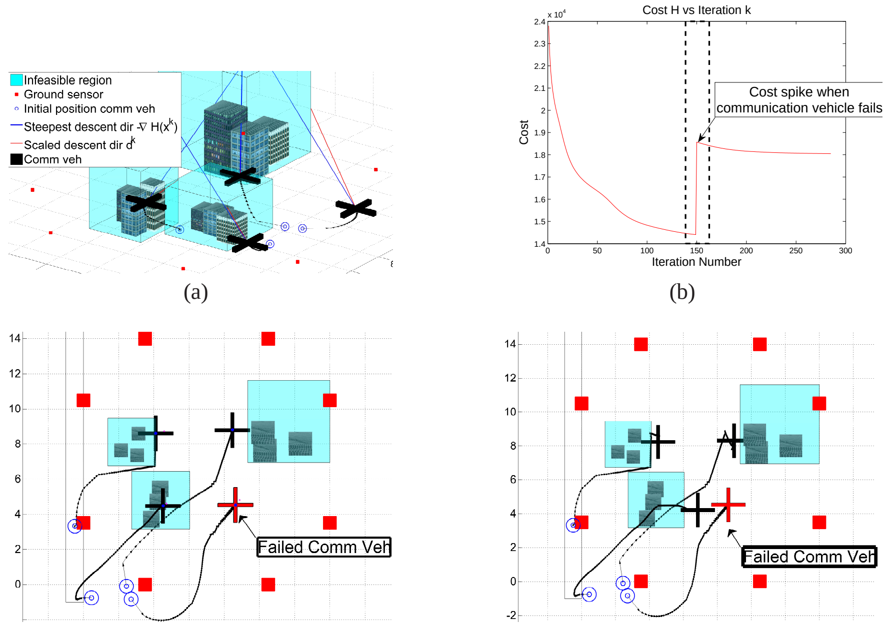
In this section we summarize our control method in Algorithm 1 and suggest a heuristic method for choosing an appropriate scaling matrix S^k for each vehicle. We demonstrate our algorithm and the suggested method for finding S^k via a Matlab simulation for four communication vehicles and eight ground sensor vehicles in three-dimensional space (Figure 3).

Algorithm 1 Decentralized Control for Optimized Comms (for agent i)

$x^k = x^0$, $k = 0$.
while $k == 0$ OR $\left|\mathbf{x^{k+1}} - \mathbf{x^k}\right| \geq tol$ **do**
  $k \leftarrow k+1$
  {Compute scaling S^k using environment topology, see Algorithm 2.}
  {Compute gradient using neighbors of agent i:} $\nabla_i H(x^k)$
  {Compute desired waypoint:} $z^k \leftarrow x^k - s^k(S^k)^{-1}\nabla_i H(x^k)$
  {Compute: }$\vec{x}_S^k$ ←soln to (4)
  {Compute feasible stepsize α^k satisfying Assumption 1.}
  {Compute new point x^{k+1} for agent i using stepsize α^k:} $x^{k+1} = x^k - \alpha^k(\vec{x}_S^k - x^k)$.
end while

4.1.1 Heuristic Selection of Scaling Matrix S^k

We suggest one possible method for choosing a scaling matrix S^k for each vehicle that is easily implemented and relies solely on map topology that is *local* to each agent. We show via simulation, the performance of the resulting optimization and its adaptive capabilities in the case of agent failures. For each agent, we draw a line along the direction of steepest descent which is plotted as a blue line in Figure 3(a), call this line gL. Let $\mathcal{O}$ be the first infeasible region intersected by gL. We wish to compute S^k such that we move around $\mathcal{O}$, so we compute the projection of the intersection point onto each of the L edges of $\mathcal{O}$ and choose the point such that the chord from the current position x^k to the edge point e^* has the largest dot product $-\nabla H(x^k)'(e^* - x^k)$. This represents a direction that is as close to the direction of steepest descent as possible but that circumvents the infeasible region obstructing this direction. This chord is plotted in red for each agent whose steepest descent direction intersects an infeasible region in Figure 3(a). We use this chord to compute the first eigenvector of S^k so that $v_1 = (e^* - x^k)/\left\|(e^* - x^k)\right\|$, then v_2 and v_3 are



(a) (b) (c) (d)

Fig. 3 (a) Scenario showing infeasible regions and 4 communication vehicles and 8 ground sensors. (b-d) Adaptive behavior when one communication vehicle fails (red quadrotor): remaining vehicles change trajectories to compensate. (b) Cost is always decreased along agent trajectories.

simply any other unit vectors that are othornormal to each other and to v_1. Finally, we can set the eigenvalues $\lambda_2, \lambda_3 >> \lambda_1$ to attain a direction d^k closest to the v_1 direction, see discussion in Section 3.2.1 and Figure 3(a). We warn however that choosing the eigenvalue ratios too large will inversely effect convergence rate and thus this should not, in practice, be made larger than necessary. The matrix S^k is then computed via its eigenvectors and eigenvalues as $S^k = V\Lambda V'$ where V and Λ are defined in Section 3.2.1. If there is no infeasible region obstructing the direction gL for that vehicle, or there exists no such edge point e^* so that the dot product $-\nabla H(x^k)'(e^* - x^k) > 0$ (this is the case where no circumventing direction produces descent in the cost H), we simply set $S^k = I$, where I is the identity matrix. This algorithm is summarized in 2. For the simulation in Figure 3 we set $\lambda_1 = 1, \lambda_2 = 50, \lambda_3 = 50$ and achieved satisfactory convergence in an average of 150 iterations where each iteration took on the order of 0.7 seconds using the CPLEX for Matlab toolbox on a 2.4GHz CPU laptop.

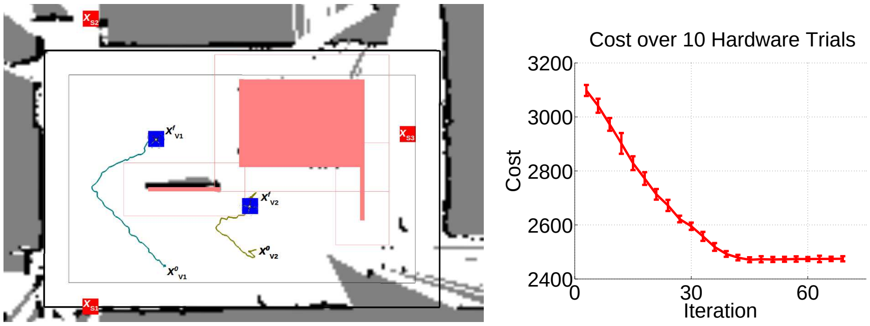
4.1.2 Discussion on When to Use Scaling

The use of scaling is most effective when applied at sufficient distance from path obstructing infeasible regions. Since any descent direction d^k must be less than perpendicular to the negative gradient direction, as a vehicle gets closer to the edge of an obstructing infeasible region, the range of descent directions that can clear the obstructing region becomes smaller. Therefore we expect scaling to perform better in environments where there are larger distances between obstacles, and where

Algorithm 2 Heuristic Selection of Scaling Matrix S^k in 3D Using Local Information (for agent i)

{Define D: Sensing radius within which infeas. regions can be detected for vehicle i.}
{Define $\mathscr{O}$: closest infeas. region in steepest descent direction. }
{Define $rot(\pi/2)$: Rotation matrix by $\pi/2$.}
{Compute a line in direction of negative grad: }$gL = x^k - D\nabla H(x^k)$.
{ $ip \leftarrow$ intersection point of gL with closest face of infeasible region ($\mathscr{O}$)}
if $ip \neq \{0\}$ **then**
  {$EP \leftarrow$ all points on edges of $\mathscr{O}$ with smallest distance from ip}
  $e^* = \max_{e \in EP}(e - x^k)' gL$
  {Compute first orthonormal eigvec of S^k:} $v_1 \leftarrow (e^* - x^k)/(\|e^* - x^k\|)$
  $v_2 \leftarrow rot(\pi/2) * v_1$
  $v_3 \leftarrow (v_1 \times v_2)/(\|v_1 \times v_2\|)$
  $V \leftarrow [v_1\ v_2\ v_3]$
  {Set $\lambda_1 << \lambda_2, \lambda_3$ as discussed in Section 3.2.1}
  $\Lambda \leftarrow diag(\lambda_1, \lambda_2, \lambda_3)$
  $S^k \leftarrow V \Lambda V^T$
else
  {No obstacles in steepest descent direction, set scaling to identity: $S^k \leftarrow I$}
end if

scaling is applied at the time that an obstructing obstacle is detected as outlined in Algorithm 2. Theorem 2 shows that as long as the scaling matrix S^k is strictly positive definite, $\bar{x}^k \neq x^k$, and x^k is not a critical point such that $\nabla H(x^k) \neq 0$, then the resulting direction d^k can never be perpendicular to the negative gradient direction. For an intuitive explanation, consider the two dimensional case and the unprojected direction $\tilde{d}^k$ from (10). As one of the eigenvectors of S^k, say v_1, becomes perpendicular to $-\nabla H(x^k)$, the component of the negative gradient in the direction of v_1 approaches zero, $\zeta_1 \to 0$ and $-\nabla H(x^k) \to \zeta_2 v_2$. Therefore the direction $\tilde{d}^k = s^k(\frac{1}{\lambda_1}\zeta_1 v_1 + \frac{1}{\lambda_2}\zeta_2 v_2) \to s^k \frac{1}{\lambda_2}\zeta_2 v_2$ which is exactly the negative gradient direction scaled by $s^k \frac{1}{\lambda_2}$. This means that even if scaling is applied incorrectly (almost perpendicular to the negative gradient), the resulting direction can never be perpendicular and in fact will align with the negative gradient direction, although, if λ_2 is a very large number it is seen that progress along this direction becomes very slow and convergence rate suffers as discussed in Section 3.2.1. Also if the current position is at a stationary point where the projection $\bar{x}^k$ is equal to x^k which may occur at the side of an obstacle, or at a critical point of the cost where $-\nabla H(x^k) = 0$, the resulting direction is zero even if nonzero scaling is applied. This can be seen easily from the update equation $x^{k+1} = x^k + \alpha^k d^k$ where $d^k = (\bar{x}^k - x^k)$ which is zero if x^k is stationary, or in free space $d^k = -s^k(S^k)^{-1}\nabla H(x^k) = 0$ at a critical point where $\nabla H(x^k) = 0$. Therefore the observations that 1) d^k can never be perpendicular to the direction of steepest descent (and actually approaches the steepest descent direction if scaling is applied perpendicular to the negative gradient), and 2) that the direction d^k is zero such that the method stops at stationary points or critical points even for positive scaling $S^k \neq I$, and finally that 3) scaling is more effective when applied at larger distances from path obstructing infeasible regions, motivate our recommendation of applying scaling for any path obstructing infeasible region within the vehicle sensing radius.



(a) (b)

Fig. 4 (a) Overhead view of field test scenario ($\sim 11m \times 7.5m$). Obstacles (pink) and configuration space boundaries (solid lines) overlay a gridmap of the environment. Stationary ground sensors (x_{S1}, x_{S2}, x_{S3}) are shown as red squares. The 2 quadrotor trajectories are shown in teal and yellow, with initial positions (x^0_{V1}, x^0_{V2}) and final positions (x^f_{V1}, x^f_{V2}) highlighted by blue squares. (b) Average trial cost and standard deviation averaged over 10 trials.

4.2 Hardware Experiments

The algorithm was validated in a decentralized hardware experiment with two mobile quadrotor helicopters and three stationary ground sensors. This evaluation was performed in a known GPS-denied indoor environment with obstacles (the second floor atrium in the Stata Center at MIT). The hardware platform consisted of Ascending Technologies Pelican quadrotors[1], each outfitted with a Hokuyo[2] UTM-30LX laser range-finder and 1.6*Ghz* Intel Atom processor (for details see [17]). Each vehicle performs onboard state estimation and control enabling completely autonomous flight. For practical purposes, each quadrotor communicates via WiFi with a corresponding ground station laptop, where human input and planning processes are run. The communication channel between the mobile and ground sensors is simulated. The environment and vehicles are shown in Figure 1.

Ten trials were run, each starting at the initial configuration shown in Figure 4 (labeled x^0_{V1}, x^0_{V2} for vehicles 1 and 2, respectively). The obstacle positions are overlayed on the gridmap in pink, and a solid outline denotes the configuration space boundaries, or *infeasible* regions. These regions do not impede communication; rather, they represent unsafe or untraversable regions. In this environment these obstacles were an open staircase, a thin wall, and a table. The quadrotors share real-time pose information and at each control iteration 10*Hz* compute their next waypoint according to Algorithm 1. The control commands were artificially throttled at 1*Hz* by the waypoint executor. Figure 4 shows the trajectory of each vehicle during one trial, and the resulting local minima configuration to which they converge. Note that vehicle 1 moved around the wall. Vehicle 2 initially moved towards the wall, then converged to a point along the obstacle boundary distributed between sensors 1 and 3 and vehicle 1. The average duration over all trials was 65*s* until convergence.

Video footage: `http://people.csail.mit.edu/prentice/isrr2011/`

[1] Ascending Technologies GmbH. `http://www.asctec.de`

[2] Hokuyo UTM-30LX Laser. `http://www.hokuyo-aut.jp`

5 Discussion

We have presented a method for communication optimization in a heterogeneous network of aerial and ground vehicles in an environment with infeasible regions using the communication cost function from previous work [5]. We pursue extension to the general nonsmooth case, and study of the effect of obstacles on communication strength in future work. We have demonstrated both analytically and through simulation and hardware experiments, the utility of using a sequence of scaling matrices to improve the range of converged solutions by moving along trajectories that avoid infeasible regions.

Acknowledgements

The authors acknowledge the MAST Project under ARL Grant W911NF-08-2-0004, the SMART Future Urban Mobility Project, and the NSFGRFP.

References

1. Ali Jadbabaie, Jie Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48:988–1001, 2003.
2. Jorge Cortes, Sonia Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20:243–255, 2004.
3. Sonia Martinez, Jorge Cortes, and F. Bullo. Motion coordination with distributed information. *Control Systems Magazine, IEEE*, 2007.
4. M Schwager, B Julian, and D Rus. Optimal coverage for multiple hovering robots with downward-facing cameras. In *Proc. ICRA*, 2009.
5. Stephanie Gil, Mac Schwager, Brian Julian, and Daniela Rus. Optimizing communication in air-ground robot networks using decentralized control. In *Proc. ICRA*, 2010.
6. M. Schwager, J. J. Slotine, and D. Rus. Unifying geometric, probabilistic, and potential field approaches to multi-robot coverage control. In *Proc. ISRR*, 2009.
7. Dimitri Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2008.
8. Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5, 1986.
9. C.W. Warren. Multiple robot path coordination using artificial potential fields. *Robotics and Automation, IEEE Internal Conference on*, 1990.
10. S. S. Ge and Y. J. Cui. Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13:207–222, 2002.
11. N. Ayanian and V. Kumar. Decentralized feedback controllers for multiagent teams in environments with obstacles. *Robotics, IEEE Transactions on*, 26, 2010.
12. J. Cores A. Ganguli and F. Bullo. Distributed deployment of asynchronous guards in art galleries. In *Proc. ACC*, 2006.
13. N. Motee, A. Jadbabaie, and G. Pappas. Path planning for multiple robots: An alternative duality approach. In *American Control Conference (ACC), 2010*, 2010.
14. C. Caicedo and M. Zefran. A coverage algorithm for a class of non-convex regions. In *Proc. CDC*, 2008.
15. E Stump, A Jadbabaie, and V Kumar. Connectivity management in mobile robot teams. In *Robotics and Automation, IEEE International Conference on*, pages 1525–1530, 2008.
16. Tom Schouwenaars, Bart DeMoor, Eric Feron, and Jonathan How. Mixed integer programming for multi-vehicle path planning. In *European Control Conference*, 2001.
17. A. Bachrach, A. de Winter, R. He, G. Hemann, S. Prentice, and N. Roy. RANGE - Robust autonomous navigation in gps-denied environments. In *Proc. ICRA*, 2010.