Constrained Multi-Agent Optimization with Unbounded Information Delay

Stefan Heid, Arunselvan Ramaswamy, Eyke Hüllermeier

Heinz Nixdorf Institute and Department of Computer Science
Paderborn University
Warburger Straße 100, 33098 Paderborn
E-Mail: {stefan.heid, arunselvan.ramaswamy, eyke}@upb.de

Abstract

A multi-agent system (MAS) consists of a group of agents that solve a common task through cooperation. Many problems arising in this setting can be formulated as distributed constrained optimization. In recent work, we considered the unconstrained version of the problem. In particular, we developed a theory to understand distributed gradient-based optimization methods, wherein the local (state) information is communicated via a lossy wireless network. A key contribution of the theory is that the information delay could be unbounded, however, it does not consider constraints. In this work, we present preliminary experimental results aimed towards extending the aforementioned work to the *constrained* setting. First, the constrained optimization problem is transformed into an unconstrained one using the penalty-based method. Then, we employ the distributed gradient approach from our previous work to solve the unconstrained optimization in a decentralized manner. The illustrative experiments are based on autonomous pattern formation tasks for robotic swarms. The (simulated) robots cooperate to form a specified pattern (line, circle), with the constraint that the distances between neighboring robots equal a given constant.

1 Introduction

A multi-agent system (MAS) is typically large-scale in nature, and a wireless communication network is used to connect the various agents involved, due to its convenience and cost. Examples of MAS include wireless sensor networks and smart grids, see [3]. Many problems that arise in these systems can be cast as constrained optimization problems that need to be solved in a distributed decentralized manner [6]. For example, in smart grids, a group of controllers has a common objective to minimize the control errors in terms of AC frequency or to maintain voltage levels in the whole grid with time-variant loads or energy sources. The controllers cooperate to solve this problem under constraints on the system state.

The literature on distributed algorithms to solve constrained optimization problems is rich, see e.g. [1]. However, they typically assume that the delay associated with the transfer of information from one agent to the other is bounded. Failed transmissions and channel delay are two main factors that contribute to information delay. In this paper, we focus on *information delay due to failed transmissions*. We study the effect of unbounded information delay on distributed algorithms for constrained optimization. In the past, unbounded information (update) delays were studied within the setting of unconstrained optimization in [4, 5].

The global objective is formulated in terms of a differentiable function. The agents solve this objective, together, by searching appropriate local subspaces via gradient steps. The solution to the global problem is obtained by putting together the distributed solutions. For local gradient calculations, at every step, the agents require information from other agents. Furthermore, each agent has to optimize subject to some local constraints. To this end, we use the penalty method to transform the constrained problem into an unconstrained one. In other words, the distributed gradient updates of each agent is augmented by a penalty term that encodes the violation of local constraints. It may be noted that the associated penalty hyper-parameter is increased over time. Since the communication channel is lossy, the information from the peers may be delayed and the agent is therefore forced to carry out update steps using *outdated information*. In [4], mild requirements on the quality of the wireless network are presented, which ensure that using outdated information does not hinder convergence. In this paper, we conjecture and present preliminary numerical results which suggest that *similar conclusions can be drawn even in the presence of constraints*.

To illustrate the ideas, we consider pattern forming tasks for robot swarms as an application. To this end, the specified pattern (line, circle) is expressed as an objective function. The objective is constructed, such that the minimum of the objective is reached when the robots arrange in the pattern. The objective function is evaluated using all robot positions, and the distances to neighboring agents constitute the local constraint set for every agent. At every time step, each robot moves in accordance to the local gradient update. To calculate this gradient, it uses the last known position of the other robots in the swarm. Since robot positions are communicated using lossy channels, the last known position may be outdated. In our experiments, we assume that the robots are ordered and communicate their knowledge of the swarm, only with direct neighbors in the chain.

2 Problem Definition

Broadly speaking, we have m agents that aim to minimize a given *global* objective function while satisfying *local* constraints. In other words, the agents cooperate to find:

$$\begin{aligned} \mathbf{z}^* &= \arg\min_{\mathbf{z}} \mathbb{E}_\xi \left[J(\mathbf{z}, \xi) \right], \qquad (1) \\ \text{s.t. } G_i &= \{ g_{ik} \leq 0 \mid 1 \leq k \leq k_i \} \quad 1 \leq i \leq m, \end{aligned}$$

where $\mathbf{z}^* = (\mathbf{z}_1^*, \ldots, \mathbf{z}_m^*)$ such that $\mathbf{z}_i^*$ is the component of the minimum that is calculated by the i^{th} agent a_i, $G_i = \{ g_{ik} \mid 1 \leq k \leq k_i \}$ is the local constraint set of a_i containing k_i inequality constraints. The stochastic objective function, $J : \mathbb{R}^n \times \mathbb{S} \to \mathbb{R}$, is such that $\mathbf{z} \doteq (\mathbf{z}_1, \ldots, \mathbf{z}_m)^T \in \mathbb{R}^n$ where $\mathbf{z}_i$ is the *local variable* associated with a_i, and ξ is an $\mathbb{S}$-valued random variable. In typical applications, $\mathbb{S}$ is some compact subset of $\mathbb{R}^k$, $k \geq 1$, or $\mathbb{R}^k$ itself. Please note that we *allow for general vector-valued* $\mathbf{z}_i$s.

The reader may note that the local constraint set G_i, of a_i, is *not visible* to a_j for $j \neq i$. In other words, the agents are only aware of their local constraints, not that of others. Since we use the penalty-based method to transform the constrained optimization problem into an unconstrained one, we may associate each G_i with the following penalty function:

$$P_i(\mathbf{z}) \doteq \sum_{k=1}^{k_i} \max\left(0, g_{ik}(\mathbf{z})\right)^2. \tag{2}$$

2.1 Communication Model

As stated earlier, the agents are connected using a wireless communication network. We model this using a weighted directed graph $G = (V, E)$. In this graph, each agent is represented as a node and a directed edge $e_{ij} = (a_i, a_j)$ exists if a_i can directly transmit messages to a_j, possibly using a dedicated *unidirectional channel*. The edge-weights ($\in [0,1]$) represent the probability of successful transmission along that edge. We assume that the transmissions along different edges are independent, i.e., there is *zero interference*. We allow for graph evolution, provided it is connected at all times. In particular, at any point in time, there exists a path connecting a_i to a_j such that the product of the edge-weights (success probabilities) is strictly greater than zero, $1 \leq i, j \leq m$. Hence, there is a chance that the message sent by a_i reaches a_j.

To find a solution, $\mathbf{z}^* = (\mathbf{z}_1^*, \ldots, \mathbf{z}_m^*)$, to the above described constrained optimization problem, a_i searches for $\mathbf{z}_i^*$ in its local search space $\mathbb{R}^{n_i}$ using the following gradient formula:

$$\frac{\partial J}{\partial \mathbf{z}_i} + \beta \frac{\partial P_i}{\partial \mathbf{z}_i}, \tag{3}$$

where $\frac{\partial}{\partial \mathbf{z}_i}$ is the partial derivative with respect to the variable $\mathbf{z}_i$, β is the *penalty parameter*, and $\sum_{i=1}^{m} n_i = n$. In order to calculate $\frac{\partial J}{\partial \mathbf{z}_i}$, a_i requires updates from a_j, $j \neq i$. This information is exchanged using the underlying wireless communication network, which causes delays. In this paper, we consider the following sources of *information delays*:

- packet losses;
- routing through other agents in the system, due to the lack of direct connection.

Note that we do not consider channel delays in this paper. However, we believe that our ideas may be readily extended to incorporate, possibly unbounded, channel delays. The delays directly affect the *age of the information* available to an agent. In this paper, we use the term *information delay* and *age of information*, interchangeably.

The gradient calculation in (3) *deals with information delays, by using the latest available updates from other agents in the system.*

2.1.1 On unbounded information delays

Let us suppose that there are no packet losses. The delay due to indirect routing grows linearly as a function of the distance between the nodes in the graph. We assume that the diameters (maximum distance between any pair of nodes) of the evolving graphs are bounded, independent of time. Hence the delay due to indirect routing is also bounded. If we now consider packet loss, then updates within any *bounded time-frame* cannot be guaranteed. Hence, *packet loss* is the major contributor to information delay. The probability that a_i successfully communicates with a_j within any d time-step interval is some $p > 0$, where d is the above mentioned bound on the graph diameter. Note that p may vary over time. Hence, the event of unsuccessful communication over successive d length intervals is geometrically distributed. In other words, there is no absolute bound on the information delay.

3 Algorithm

We are now ready to present an algorithm to solve (1). It may be noted that it is based on penalty-based gradient descent methods for the centralized version. In the setting considered here, agent a_i updates $\mathbf{z}_i$ in an iterative manner, through gradients calculated using the latest available $\mathbf{z}_j$, $j \neq i$. At any time t, a_i

Algorithm 1.: Distributed Optimization

1: Initialize a_i with $\mathbf{z}$
2: **for all** time-step **do**
3: process received $\hat{\mathbf{z}}_j^t$
4: update $\mathbf{z}_i^t$
5: **for all** $a_j : (a_i, a_j) \in E$ **do**
6: send $\hat{\mathbf{z}}_i^t$ to a_j

maintains a *local view*, $\hat{\mathbf{z}}_i^t$, of the global variable $\mathbf{z}^t$. Formally speaking, the local view of a_i at time t is given by $\hat{\mathbf{z}}_i^t \doteq \left(\mathbf{z}_1^{\tau_{i1}(t)}, \ldots, \mathbf{z}_i^t, \ldots, \mathbf{z}_m^{\tau_{im}(t)}\right)^T$, where $0 \leq \tau_{ij}(t) \leq t$, and $t - \tau_{ij}(t)$ is the *age of the information* from a_j available to a_i at time t. The local $\mathbf{z}_i$ is updated as follows:

$$\mathbf{z}_i^{t+1} \leftarrow \mathbf{z}_i^t - \eta(t)\left[\frac{\partial J(\hat{\mathbf{z}}^t, \xi^t)}{\partial \mathbf{z}_i} + \beta(t)\frac{\partial P_i(\hat{\mathbf{z}}^t)}{\partial \mathbf{z}_i}\right], \tag{4}$$

where $\eta(t)$ is the learning rate and $\beta(t)$ is the time-varying penalty parameter. ξ^t are statistically independent samples that have the same distribution as ξ.

3.1 Information exchange

At time t, a_i sends $\hat{z}_i^t$ to its neighbors. Simultaneously, it receives $\hat{z}_j^t$s from a subset of its neighbors (some may be lost due to packet drops). It uses the obtained information, and z_i^t, to update $\hat{z}_i^t$ to $\hat{z}_i^{t+1}$, such that it contains the latest variables associated with other agents. To facilitate consistent updates, we assume that each variable is associated with a *time stamp*. Hence, at time-step t, a_i receives $\hat{\mathbf{z}}_j^t$ along with the vector of time stamps $\hat{\mathbf{t}}_i \doteq (\tau_{j1}(t), \ldots, t, \ldots, \tau_{jm}(t))^T$, from a subset of its neighbors. Using the obtained information, a_i updates the entries of $\hat{\mathbf{z}}_i^t$ by comparing time stamps. In other words, a_i checks to see if $\tau_{jk}(t) > \tau_{ik}(t)$, if yes, then updates $\hat{\mathbf{z}}_i^t(k)$ to $\hat{\mathbf{z}}_j^t(k)$. It also updates the corresponding entry in $\hat{\mathbf{t}}_i$. Otherwise, old entries are retained. Note that $\hat{\mathbf{z}}_i^t(k)$ is used to represent the information that a_i has of a_k, at time t. Subsequently, a_i executes update step (4). Finally, the agent sends its updated $\hat{\mathbf{z}}_i^{t+1}$ along with the updated time stamps to all its neighbors. This allows, agent a_j similarly to discard outdated updates. The reader must note that no retransmissions are triggered in case of failed transmissions since we do not assume that received packets are acknowledged.

The discussion in this section has been codified in Algorithm 1.

3.2 On the convergence of Algorithm 1

We believe that a proof of convergence of the algorithm will proceed along similar lines as the analysis in [4]. Here, we studied the unconstrained version of Algorithm 1. Suppose the random variables associated with the age of information, at every time-step, have bounded second moments, then it is shown, in [4], that the associated errors are asymptotically in the order of the learning rate. Since the learning rate diminishes to zero, the effect of information delays vanishes asymptotically. Also, that the distributed algorithm has the same asymptotic properties of a centralized one.

A sufficient condition on the wireless network to ensure the above mentioned bounded second moment requirement is stated as assumption (A6) in [4]. It is restated below for our setting:

- for each pair of agents, there is a non-zero probability of successful transmission of the routed package,
- and the transmission probabilities of all edges are statistically independent.

As discussed in Section 2.1, the communication graph is always connected. Hence the above statement conditions are readily satisfied.

Now, we discuss the influence of the penalty parameters on convergence. In our algorithm, we do not use a constant β, rather we take $\beta(t) \uparrow \infty$. This is done to avoid the scenario wherein the algorithm converges to $\mathbf{z}^\infty$ such that:

$$\begin{aligned} \nabla J(\mathbf{z}^\infty) + \beta \sum_{i=1}^{m} \nabla P_i(\mathbf{z}^\infty) &= 0 \text{ and} \\ \nabla J(\mathbf{z}^\infty) &= -\beta \sum_{i=1}^{m} \nabla P_i(\mathbf{z}^\infty) \neq 0. \end{aligned}$$

This phenomenon is explained in the literature of centralized penalty-based method. As stated earlier, we can use the arguments from [4] to conclude that Algorithm 1 has the same long-term behavior as its centralized counterpart. In other words, $\beta(t) \uparrow \infty$ is important to ensure that $\nabla J(\mathbf{z}^\infty) = \sum_{i=1}^{m} \nabla P_i(\mathbf{z}^\infty) = 0$, as desired.

However, the main issue with a time-varying penalty parameter that goes to infinity is the growth in the *variance of the descent directions*. Hence, a diminishing learning rate is required to counteract this. More precisely, $\eta(t)$ and $\beta(t)$ are chosen such that

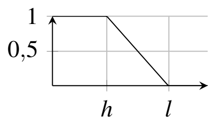
$$\sum_t \big(\eta(t) \cdot \beta(t)\big)^2 < \infty.$$

This condition is inspired by a similar assumption, A1, in [7]. Intuitively, it is clear that a condition such as $\eta(t)\beta(t) \to 0$ is required. However, it is shown in [7] that is not always sufficient.

4 Experiments

In this section, we present the results of two experimental studies, in which mobile robots are simulated as points in the two-dimensional Euclidean space[1]. The two scenarios optimize for a different objective function and have slightly different penalties, while the communication model is the same for all experiments. A simple packet reception probability model is used to calculate the success probability of transmission as a function of robot distance. To ensure connectedness of the communication graph, the minimal transmission probability is bounded as follows:

$$p_{h,l}(d) = \mathbf{clip}\left(\frac{h-d}{h-l}, 0.01, 1\right)$$



[1] https://github.com/stheid/DDSCO

The parameters h and l determine the thresholds of maximum and minimum transmission probability, respectively.

The general setting of the following scenarios are robot-swarm pattern formation tasks. Each agent is represented by a point $\mathbf{z}_i = (x_i, y_i)^T \in \mathbb{R}^2$. In the first scenario, the agent's objective is to form a line on which the agents evenly space out. In the second scenario, the agents should form a circle and also create a constant distance to each other. The inter-robot distance is modeled by the constraints, while the created structure is modeled in the objective function. Initially, the agents are placed uniformly at random in a quadratic region. The vector $\mathbf{z}$ is the concatenation of local position vectors $\mathbf{z}_i$ of all agents a_i. Additionally, we denote $\mathbf{x} = (x_1, \ldots, x_m)^T$ and $\mathbf{y} = (y_1, \ldots, y_m)^T$. In our experiments, we simulate a swarm of $m = 10$ robots.

4.1 Scenario 1: Forming a Line

The objective function for forming a line consists of two parts. The first component is the residual error of the ordinary least squares (OLS) regression over all points. The second part is the distance between the first and last point. Maximizing the distance along with minimizing the residual error encourages the robots to unravel if they form a folded line.

The x-term of each position is augmented by a constant for fitting the bias term and a random value to make the objective more challenging: $\phi(x_i) = (1, \gamma_i, x_i)$ with $\gamma_i \sim \mathcal{N}(\mu = 0, \sigma = 0.1)$. For ease of notation we define $\Phi = (\phi(x_1)^T, \ldots, \phi(x_m)^T)^T$ as the transformation of $\mathbf{x}$. Formally the objective is defined as follows:

$$
\begin{aligned}
J(\mathbf{x}, \mathbf{y}) &= \frac{\mathbf{e}^T\mathbf{e}}{n^2} - \frac{\|\mathbf{z}_1 - \mathbf{z}_m\|^2}{n} \\
\mathbf{e} &= \mathbf{y} - \Phi\mathbf{b} \\
\mathbf{b} &= (\Phi^T\Phi)^{-1}\Phi^T\mathbf{y}
\end{aligned}
$$

Here, $\mathbf{b}$ is the OLS regression line and $\mathbf{e}$ is the residual error of the regression estimate. Since the regression error is in the order of the magnitude of the squared number of nodes, it has been normalized accordingly. Similarly, the distance between the first and last agent of a chain is normalized by the number of agents.

Each agent has up to two local constraints. The first and last agent have only one neighbor, therefore, they will only have one constraint. All other agents have two equality constraints to maintain a constant distance to their neighbors.

More precisely, agent i has the following constraints:

$$g_{i1}(\mathbf{z}) = |d_0 - \|\mathbf{z}_i - \mathbf{z}_{i-1}\|^2| = 0 \qquad \text{if } i > 1$$
$$g_{i2}(\mathbf{z}) = |d_0 - \|\mathbf{z}_i - \mathbf{z}_{i+1}\|^2| = 0 \qquad \text{if } i < m$$

with d_0 the demanded inter agent distance and $\|\cdot\|^2$ the Euclidean distance.

4.2 Scenario 2: Forming a Circle

Several objective functions could be chosen to form a circle. A quite obvious one would be to make the agents maximize the area of the polygon they span. Correctly calculating the area of arbitrary polygons is a non-trivial task, as self-intersecting polygons pose additional challenges [2]. Fortunately, the equation for calculating the area of simple polygons can be used as a lower bound for the actually covered area. Therefore, it is sufficient to use this equation for arbitrary polygons in our case, since we aim for maximization:

$$J(\mathbf{x},\mathbf{y}) = -\left|\sum_{i=0}^{n-1} x_i y_{i+1} - x_{i+1} y_i\right| \qquad \text{with } x_0 = x_m \text{ and } y_0 = y_m$$

For simplicity, the constant prefix has been omitted, as it will not influence the position of a minimum. Since the algorithm is purely guided by the gradient, it might converge to local optima.

The constraints are similar to the previous example, however, the first and last agent will now be considered as neighbours:

$$
\begin{aligned}
g_{i1}(\mathbf{z}) &= \left| d_0 - \|\mathbf{z}_i - \mathbf{z}_{i-1}\|^2 \right| = 0 \\
g_{i2}(\mathbf{z}) &= \left| d_0 - \|\mathbf{z}_i - \mathbf{z}_{i+1}\|^2 \right| = 0 \qquad\qquad \text{with } \mathbf{z}_0 = \mathbf{z}_m
\end{aligned}
$$

The hyperparameters for learning rate and penalty scale are chosen as follows:

$$
\begin{aligned}
\eta(t) &= \frac{100}{t + 30000} \\
\beta(t) &= 1 + \sqrt{\frac{t}{100}}
\end{aligned}
$$

The learning rate is therefore decreasing in a inverse proportional manner, while the the penalty term goes to infinity in the square root of the timestep. The constants in the equation were tuned by hand to achieve fast convergence.

4.3 Results

In both of the above experimental scenarios, we observed convergence to at least a local minimum, while satisfying all local constraints. For the line scenario, the robots eventually arrange on a line, however, sometimes the line is folded into itself, which will not maximize the distance, but minimize the regression error. For the circle scenario, we similarly see the forming of perfect circles or shapes like spirally intersecting circles, which represent a local maximum of the covered area.

The rate of convergence was observed to strongly depend on the quality of the wireless network (transmission success probabilities). Figure 1 illustrates the evolution of the algorithm, tasked to form the circle, in the second scenario. In the beginning, the agents were severely penalized due to large inter-agent distances. This causes them to move towards each other, and form a closely-knit cluster. In the next phase, the agents try to form a larger circle, due to the design of the objective function. In the final phase, the increasing penalty

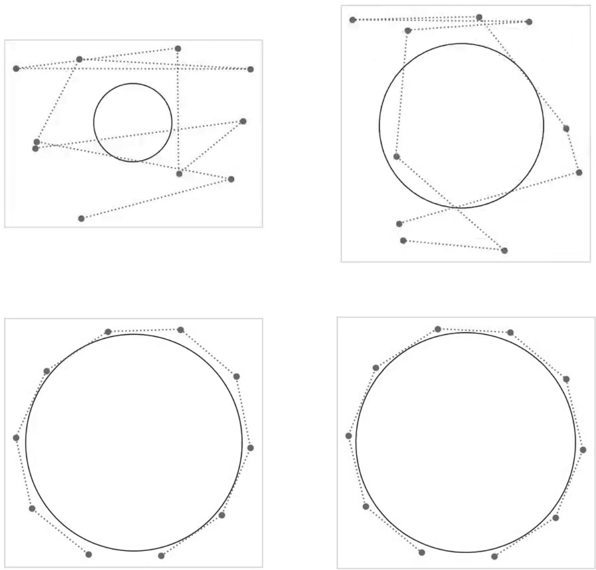


Figure 1: Snapshots of different phases of scenario 2. Each dot represents the position of a robot, the communication link is visualized as the blue line. The optimal solution is indicated with the black circle.

parameter forces the agents to *move closer to each other*, to fulfill the distance constraint.

4.3.1 Impact of Communication Quality on Rate of Convergence

The quality of the wireless network seems to affect the rate of convergence. To illustrate this, we experimented with different success probabilities (network qualities). Empirical results suggest that a strong correlation cannot be directly seen. This is because aged information about the peer's positions allow constraints to be violated more freely in some cases. In other words, bad communication may allow for a streamlined convergence to a wrong minimum (for e.g., does not satisfy constraints). Loosely speaking, old information facilitates

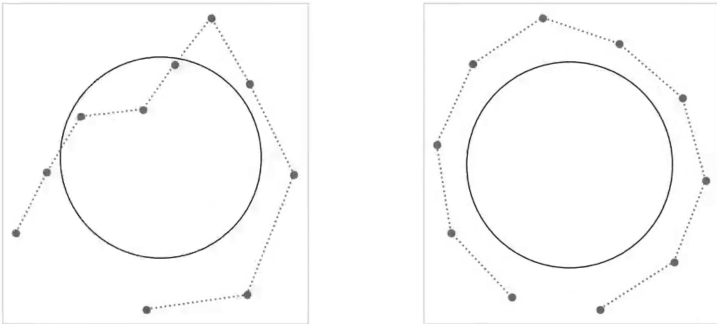


Figure 2: Convergence progression in scenario 2 after about 2000 update steps. Left picture shows convergence with less reliable communication channels and hence slower convergence.

a more localized optimization and constraint satisfaction, and updated information brings back a clearer picture of the global constrained optimization. Figure 2 shows the convergence of two identical configurations of scenario 2, which only vary in the communication channel quality. The optimal solution subject to the distance constraints is reached when all agents arrange on the black circle, in an equally spaced manner. The figure to the right illustrates the scenario with good communication, and the figure on the left the bad one. The algorithm has almost formed a circle, under good communication, only some constraint violations are left to be addressed. In the case of bad communication, the algorithm has not yet formed a circle and is hence lagging.

4.3.2 Constraint Satisfaction

Let us consider the first scenario, wherein the agents try to form a straight line while maximizing the spanned distance of the collection. However, the constraint requires that a given inter-agent distance be achieved. With small penalty scaling factors, the constraint is very loose and can easily be violated. Therefore, the penalty needs to be scaled continuously to allow for arbitrarily small constraint violations. The continuous increase of penalty allows a smooth transition from the unconstrained to the fully constrained scenario. Similarly, in the second scenario, the agents aim to form a circle of area. Therefore, drifting outwards, however, again inter-agent distances must be maintained. Again, the penalty term needs to grow to infinity, to dominate the overall penalized objective when the solution does not satisfy constraints. In particular, growing the penalty parameter prevents convergence to a point that satisfies $\nabla f = -\beta \nabla \sum_i P_i \neq 0$.

The experimental results are in agreement with the general argument of convergence stated at the end of Section 3.

Videos showing the evolution of the convergence can be found in the repository https://github.com/stheid/DDSCO.

5 Conclusion

We considered the problem of distributed constrained optimization with stochastic objective and inequality-type constraints. To solve this problem, we presented a penalty-based distributed gradient algorithm. We presented preliminary empirical results to support the conjecture that results from [4] naturally extend to the inclusion of constraints. In particular, that the convergence, in the presence of local constraints, is unaffected by stochastic information delays with bounded second moments.

For visualization, we investigated two scenarios of pattern formation in robot swarms. The objective function was used to specify the pattern, subject to the inter-robot distance constraints. The agents collectively minimized the objective function by searching in appropriate subspaces. In the experiments, we observed that the convergence speed correlates with the quality of the communication channel, although a more thorough investigation is needed to paint a clearer picture. Also, we look forward to analyzing the setting in a more formal way to prove the convergence theoretically.

References

[1] Stephen Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. 3(1):1–122.

[2] Bart Braden. The surveyor's area formula. *The College Mathematics Journal*, 17(4):326–337, 1986.

[3] Chun-Hao Lo and Nirwan Ansari. Decentralized controls and communications for autonomous distribution networks in smart grid. 4(1):66–77. Conference Name: IEEE Transactions on Smart Grid.

[4] Arunselvan Ramaswamy, Adrian Redder, and Daniel E. Quevedo. Optimization over time-varying networks with unbounded delays.

[5] Benjamin Sirb and Xiaojing Ye. Decentralized consensus algorithm with delayed and stochastic gradients. 28(2):1232–1254.

[6] Wen-Zhan Song, Renjie Huang, Mingsen Xu, Andy Ma, Behrooz Shirazi, and R. Lahusen. Air-dropped sensor network for real-time high-fidelity volcano monitoring. pages 305–318, 01 2009.

[7] J. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37:332–341, 1992.