On Local Computation for Optimization in Multi-Agent Systems

Robin Brown[1], Federico Rossi[2], Kiril Solovey[1], Matthew Tsao[1], Michael T. Wolf[2], and Marco Pavone[1]

***Abstract*—A number of prototypical optimization problems in multi-agent systems (e.g., task allocation and network load-sharing) exhibit a highly local structure: that is, each agent's decision variables are only directly coupled to few other agent's variables through the objective function or the constraints. Nevertheless, existing algorithms for distributed optimization generally do not exploit the locality structure of the problem, requiring all agents to compute or exchange the full set of decision variables. In this paper, we develop a rigorous notion of "locality" that quantifies the degree to which agents can compute their portion of the global solution based solely on information in their local neighborhood. This notion provides a theoretical basis for a rather simple algorithm in which agents individually solve a truncated sub-problem of the global problem, where the size of the sub-problem used depends on the locality of the problem, and the desired accuracy. Numerical results show that the proposed theoretical bounds are remarkably tight for well-conditioned problems.**

I. Introduction

Many problems in multi-agent control are naturally posed as large-scale optimization problems, where knowledge of the problem cost function and constraints is distributed among agents, and the collective actions of the network are summarized by a global decision variable. Concerns about communication overhead, privacy, and robustness in such settings have motivated the need for distributed solution algorithms that avoid explicitly gathering all of the problem data in one location. This is often abstracted as a prominent setting in the literature on distributed optimization where the objective function is the sum of privately known functions, and agents must reach a *consensus* on the optimal decision variable despite limited inter-agent communication. We refer the reader to [1] for a recent survey on distributed optimization.

For many practical settings, seeking consensus as the end goal accurately represents the objective; for instance, in rendezvous and flocking problems, all the agents' actions depend on a global decision variable (meeting time and location for the former, and speed and heading for the latter). However, when the global decision variable represents a concatenation of individual actions, the network can still act optimally without ever coming to a consensus. Consider, for example, a task allocation problem where each agent only needs to know what tasks are assigned to itself, and is not concerned with other agents' assignments.

[1]R. Brown, K. Solovey, M. Tsao, and M. Pavone are with the Autonomous Systems Laboratory, Stanford University, Stanford, CA, 94305, {rabrown1, kirilsol, mwtsao, pavone}@stanford.edu.

[2]F. Rossi, and M. T. Wolf are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109, {federico.rossi, michael.t.wolf}@jpl.nasa.gov.

Part of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Many existing distributed optimization algorithms leverage consensus as a core building block and, broadly speaking, can be abstracted as the interleaving of descent steps, to drive the solution to the optimum, and averaging of information from neighbors, to enforce consistency. The main features differentiating these algorithms from each other are the centralized algorithm from which they are derived, and details regarding the communication structure such as synchronous or asynchronous, and directed or undirected communication links, with the broad overarching categories being consensus-based (sub)gradient ([2], [3]), (sub)gradient push ([4], [5]), dual-averaging ([6], [7]), and second-order schemes ([8], [9]).

Historically, the mixing time of the communication graph has been seen as a fundamental limit on the convergence of distributed optimization algorithms [7]. Accordingly, a large body of the literature focuses on designing gossip matrices whose spectral properties allow for faster mixing of information [10], [11]. This perspective implicitly makes the assumption that convergence cannot be achieved until problem information has been disseminated and subsequently incorporated into the estimates of all of the agents. Our objective in this paper is to identify problems where this global mixing is an unnecessary overhead, by quantifying how well agents can compute their portion of the global solution based solely on information in their local neighborhood.

Our approach builds on the work of Rebeschini and Tatikonda [12], who introduced a notion of "correlation" among variables in network optimization problems. The authors in [12] characterize the "locality" of network-flow problems, and show that the notion of locality can be applied to develop computationally-efficient algorithms for "warm-start" optimization, i.e., re-optimizing after the problem is perturbed.

Our approach in this paper also draws influence from the field of local computation, a sub-field of theoretical computer science. Motivated by the common threads in problems such as locally decodable codes, and decompression algorithms, Rubinfeld et al. [13] proposed a unifying framework of Local Computation Algorithms (LCAs). LCAs formalize the intuition that, in problems with large inputs and outputs, if only a small subset of the output is needed, it is inefficient to compute the entire output and simply read off the component required. Instead, both computation and access to the input should be kept to a minimum such that the required output is obtained and is consistent with subsequent queries.

Statement of Contribution: We develop a theoretical basis for the local-computation paradigm applied to convex optimization problems in multi-agent systems. Specifically, given the objective of computing x_i^*, a single component of the optimal decision variable, we characterize the error incurred by truncating the optimization problem to a neighborhood "around" x_i. We show that for *all* linearly-constrained strongly-convex optimization problems, this error decays exponentially with the size of the neighborhood at a rate dependent on the conditioning of the problem. This rate, which we coin as the "locality" of a problem, naturally characterizes the trade-off between the amount of local knowledge available to an agents, and the quality of its approximation. The condition number of a problem, colloquially referred to as a metric of how "well-behaved" a problem is, unsurprisingly, correlated with the locality of a problem. Our findings give a theoretical basis for a rather simple algorithm, in which agents simply solve truncated sub-problems of the global problem. Our numerical results, obtained by using this algorithm, show that the tightness of the theoretical bounds also depend on the condition number of the problem, with the bounds being near-optimal for well-conditioned problems.

A preliminary version of this work was accepted at the 2020 European Control Conference [14]. This paper extends prior results by providing tighter bounds on the locality of problems, and extending the decay results to *all* linearly-constrained strongly-convex optimization problems.

Organization: In Section II, we introduce notation, terminology, and technical assumptions about the problem. In Section III, we provide the problem statement, which establishes the fundamental question of locality, and summarize the main result, which provides a problem-specific bound on the rate of locality. We also summarize the key intermediary results, and discuss the algorithmic implications of locality in terms of the communication and message complexity it implies. Proof sketches of the main results are reported in Section IV. In Section V we provide numerical experiments that highlight both scenarios where our theoretical bounds are tight, and those where our bounds are conservative. We conclude and highlight future directions in Section VI.

II. Notation and Assumptions

We let $[N]$ denote the $1-N$ indices, and e_i the canonical ith basis vector. For a matrix A, A_{ij} denotes the element in the ith row and jth column. Similarly, $A_{i,*}$ and $A_{*,j}$ denote the ith row and jth column of A respectively. Let A^T be the transpose, and A^{-1} be the inverse. Given subsets $I \subseteq M, J \subseteq N$, let $A_{I,J} \in \mathbb{R}^{|I|\times|J|}$ denotes the submatrix of A obtained by restricting A to the rows in I and columns in J. Similarly, $A_{-I,-J}$ denotes the submatrix of A obtained by removing rows I and columns J. We let $\sigma_{\max}(A)$ and $\lambda_{\max}(A)$ denote the maximum singular values and eigenvalues of A respectively ($\sigma_{\min}(A)$ and $\lambda_{\min}(A)$ the minimums), and $\kappa(A) = \frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|}$ the condition number. The difference between sets, $S_1 \setminus S_2 = \{s \in S_1 \mid s \notin S_2\}$ is the set of elements in S_1 but not S_2.

Throughout this paper, we will consider linearly-constrained convex optimization problems of the form:

$$\begin{aligned} \underset{x\in\mathbb{R}^N}{\text{minimize}} \quad & f(x) = \sum_i f_i(x_i) \\ \text{subject to} \quad & Ax = b. \end{aligned}$$

We assume that $A \in \mathbb{R}^{M\times N}$ is full rank, and that each function $f_i : \mathbb{R}^N \to \mathbb{R}$ is L-smooth, μ-strongly convex, and twice continuously differentiable. We let $V^{(p)} = [N]$ denote the set of primal variables, $V^{(d)} = [M]$ the set of dual variables, and $S_j = \{i \in V^{(p)} | A_{ji} \neq 0\}$ the set of primal variables participating in the jth constraint. For any subset of the primal variable, $S \subseteq V^{(p)}$, we also define the following set of constraints

$$C_S := \{i \in [M] \mid \text{ if } j \notin S \text{ then } A_{ij} = 0\}.$$

Intuitively, C_S is the set of constraints that *only* involve variables in S. Throughout this paper, we fix the objective function f and the constraint matrix A, and write $x^*(b)$ as a function of the constraint vector, b.

We define an undirected graph $G = (V,E)$ by its vertex set V and edge set E, where elements $(i, j) \in E$ are unordered tuples with $i, j \in V$. We define the graph distance $d_G(i, j)$ to be the length of the shortest path between vertices i and j in graph G, and $\mathcal{N}_k^G(i) = \{j \in V \mid d_G(i, j) \leq k\}$ to be the k-hop neighborhood around vertex i in graph G for a given $k \in \mathbb{N}_{>0}$. We define the following undirected graphs representing coupling in the optimization problem:

- $G_{\text{dec}} = (V^{(p)}, E_{\text{dec}}(x))$, with $E_{\text{dec}} = \{(v_i^{(p)}, v_j^{(p)}) | A_{ki} \neq 0, A_{kj} \neq 0 \text{ for some } k\}$. Informally, G_{dec} encodes the *decision variables* that appear in the same constraint.
- $G_{\text{con}} = (V^{(d)}, E_{\text{con}})$, with $E_{\text{con}} = \{(i, j) | [AA^T]_{ij} \neq 0\}$. Informally, G_{con} encodes connections between the *constraints* through shared primal variables.
- $G_{\text{opt}} = (V^{(p)} \cup V^{(d)}, E_{\text{opt}}(x))$, with $E_{\text{opt}} = \{(v_j^{(p)}, v_i^{(d)}) | A_{ij} \neq 0\}$. Informally, G_{opt} encodes the dependence structure of the *optimization problem*.

III. Foundations of Locality, and Their Algorithmic Implications

A. Problem Statement

We consider a network of N agents collectively solving the following linearly-constrained optimization problem

$$\begin{aligned} \underset{x\in\mathbb{R}^N}{\text{minimize}} \quad & f(x) = \sum_i f_i(x_i) \\ \text{subject to} \quad & Ax = b, \end{aligned} \tag{1}$$

where knowledge of the constraints is distributed, and the decision variable represents a concatenation of the decisions of individual agents. Specifically, we assume that f_j and A_{*j} are initially known by agent j only, and agent j knows b_i if $A_{ij} \neq 0$. As a motivating example, consider a scenario where a fleet of agents needs to collectively complete tasks at various locations, while minimizing the cost of completing such tasks. In this setting, the constraints ensure completion of the tasks, while the entries A_{ij} of the constraint matrix may encode the portion of task i that agent j can complete, or efficiency when completing tasks, thus, constituting private knowledge. We refer the reader to Section V for additional examples.

As a departure from much of the literature on distributed optimization, we consider the problem to be solved when each agent j knows x_j^*, i.e, we do not require every agent to know the entire decision variable. With some abuse of notation, we conflate each agent with its associated primal variable[1].

Our objective in this paper is to characterize the accuracy with which an agent i can compute its associated solution x_i component given access to problem data held by agents within a k-hop neighborhood of itself in G_{dec}, for a given $k \in \mathbb{N}_{>0}$. On the communication graph given by G_{dec}, obtaining this information requires k communication rounds of accumulating and passing problem data between neighbors. Consequently, our results also characterize the trade-off between communication and approximation accuracy in this setting. This communication graph should not be seen prescriptive, but rather one that facilitates ready analysis of the implications of locality with regard to communication.

B. Foundations of Locality

For each x_i, we consider sub-problems induced by restricting Problem (1) to variables within the k-hop neighborhood around x_i and constraints only involving those variables (the "k-hop local sub-problems"). The main result of this paper states that the error in the ith component of the k-hop "local solution" decays exponentially with the size of the neighborhood. A formal statement of this result is provided below.

Theorem III.1 (Locality)**.** Let $x^{(k)}$ be the solution to the optimization problem induced by restricting Problem 1 to k-hop neighborhood around x_i, $\mathcal{N}_k^{(\text{dec})}(i)$, and the constraints only involving those variables. If $\lambda = \sup_x \frac{\sqrt{\kappa(x)}-1}{\sqrt{\kappa(x)}+1}$, where $\kappa(x)$ denotes the condition number of $A\nabla^2 f(x)^{-1}A^T$, then

$$|x_i^{(k)} - x_i^*| \leq C\lambda^k \tag{2}$$

for $C = 2\left(1+\sqrt{\frac{L}{\mu}}\right)\frac{\sigma_{max}(A)}{\sigma_{min}^2(A)}\left\|b - Ax_{UC}^*\right\|_2$.

The rate λ characterizes the degree to which local information is sufficient to approximate individual components of the global optimum, thus justifying it as a metric of "locality". The proof of Theorem III.1 relies on two intermediary results.

Remark. The definition of C may appear slightly unusual because the term $\frac{\sigma_{max}(A)}{\sigma_{min}^2(A)}$ is not scale-invariant i.e., $\frac{\sigma_{max}(cA)}{\sigma_{min}^2(cA)} = \frac{\sigma_{max}(A)}{|c|\sigma_{min}^2(A)}$. This is remedied by the fact that any constant rescaling of A and b will rescale $\left\|b - Ax_{UC}^*\right\|_2$ as well. Consequently, for all $c \in \mathbb{R}$,

$$\frac{\sigma_{max}(A)}{\sigma_{min}^2(A)}\|b - Ax_{UC}^*\|_2 = \frac{\sigma_{max}(cA)}{\sigma_{min}^2(cA)}\|cb - cAx_{UC}^*\|_2.$$

Our first intermediary result derives the relationship between solutions to the local sub-problems and the true solution to Problem (1) (the "global problem"). Specifically, we show that the solution to a local sub-problem is consistent with that of a perturbed version of the global problem (where the perturbation appears in the constraint vector, b).

Theorem III.2 (Relationship between local sub-problems and the global problem)**.** Let $S \subseteq V^{(p)}$ be a subset of the primal variables. If $x^{(S)}$ is the solution to the problem obtained by restricting Problem (1) to the variables in S and constraints only involving those variables, i.e.,

$$\begin{aligned} x^{(S)} = \underset{x^{(S)} \in \mathbb{R}^{|S|}}{\arg\min} \quad & \sum_{i \in S} f_i\left(x_i^{(S)}\right), \\ \text{subject to} \quad & A_{C_s,S}x^{(S)} = b_{C_s}, \end{aligned} \tag{3}$$

then there exists $\hat{b} \in \mathbb{R}^M$ such that $x^{(S)} = \left[x^*(\hat{b})\right]_S$.

The importance of Theorem III.2 lies in the fact that we can interpret solving local sub-problems as solving perturbed versions of the global problem. This interpretation allows us to leverage theory on the sensitivity of optimal points of Problem (1) to characterize the error incurred by only using a subset of the original problem data.

Our second intermediary result characterizes the component-wise magnitudes of this correction factor. Specifically, we show that when the constraint vector of Problem 1 is perturbed, the impact of the perturbation decays exponentially with distance to the perturbation.

Theorem III.3 (Decay in sensitivity of optimal points)**.** Let λ be defined as in Theorem III.1. Then for any perturbation in the constraint vector, $\Delta \in \mathbb{R}^M$, subset of the primal variables, $S \subseteq V^{(p)}$, and $C = \frac{2\|\Delta\|_2}{\sigma_{\min}(A)}$,

$$\|[x^*(b+\Delta) - x^*(b)]_S\|_2 \leq C\lambda^{d(S,\text{supp}(\Delta))}.$$

Intuitively, this theorem states that a perturbation in the constraints affects the decision variables "closest" to the constraint the most, i.e., those that are actually involved in the constraint, while the effect of the perturbation decays with the degrees of separation between a decision variable and the constraint. The construction of the k-hop local sub-problems takes advantage of this theorem by forcing the "perturbation" to be at a distance of at least k from component x_i. Theorem III.1 is derived from the intermediary results by bounding the perturbations induced by cutting constraints.

C. Algorithmic Implications

The characterization of locality naturally suggests a means of reducing the communication necessary for distributed optimization. In a radical departure from much of the existing work on distributed optimization, which rely on propagating information throughout the network, we suggest *localizing* information flow. Our results show that the importance of problem data to individual solution components decays with distance to the data. Consequently, if a problem exhibits sufficient locality, by restricting information flow to where it matters most, we can avoid the high communication overhead of flooding methods with little impact on solution quality.

The objective is for each agent to compute its own component of the solution vector, i.e., for agent i to compute x_i^*. We

[1]While, in this paper, each agent is only associated with a scalar variable for illustrative purposes, one can readily extend the results in this paper to the setting where each agent is associated with a vector. Additionally, the case where multiple agents' actions depend on shared variables can be addressed by creating local copies of those variables and enforcing consistency between agents who share that variable through a coupling constraint.

denote by $\hat{x}_i$ agent i's estimate of x_i^* and we let $\hat{x} = (\hat{x}_1, \dots, \hat{x}_N)$ be the aggregation of privately known solution components. Because we allow the approximation to violate constraints, the typical metric of sub-optimality in the objective function is uninformative—the approximation generated is guaranteed to have an objective value no larger than the true optimum. Consequently, we will measure the accuracy of our solution by $\|\hat{x} - x_*\|_\infty$—this bound readily translates into bounds on both the objective value and constraint violation as well.

The locality-aware distributed optimization algorithm is conceptually simple. Leveraging locality, we conclude that each agent can compute its component of the solution by solving a local sub-problem of the global problem, where the size of the local sub-problem depends on the accuracy desired and the locality parameter of the global problem. Agents aggregate local problem data through a recursive flooding scheme, which is truncated after a predetermined number of communication rounds. Then, each agent solves its own local problem without further communication with the network. Specifically, agent i starts with its local objective function, f_i, its associated column of the constraint matrix A_{*i}, and components of the constraint vector b_{C_i}. In the initialization phase, agent i sends $A_{C_i i}$ to each of its neighbors. After the initialization phase, agent i has full knowledge of $A_{C_i *}$, i.e., the constraints that it participates in. Then, in the first iteration, agent i sends a representation of $A_{C_i *}$, b_{C_i} and f_i to each of its neighbors. In subsequent iterations, each agent sends a representation of all of the information it has previously received to each of its neighbors. After the k'th iteration, for $k \in [K]$, agent i has a representation of f_j, b_{C_j} and $A_{C_j *}$ for all $j \in \mathcal{N}(i,k)$, where $\mathcal{N}(i,k)$ denotes the k-hop neighbors of agent i. After the K communication rounds, agent i generates its local sub-problem by ignoring any constraints involving variable outside of its K-hop neighborhood, $\mathcal{N}(i,K)$. The algorithm for agent i is summarized in Algorithm 1.

Algorithm 1: Locality-Aware Distributed Optimization

input: f_i, A_{*i}, b_{C_i}, K
1 Initialization: Send $A_{C_i i}$ to all $j \in \mathcal{N}(i,1)$;
2 **for** $k = 1, \dots, K$ **do**
3 | Send $\{f_l, A_{C_l,*}, b_{C_l}\}_{l \in \mathcal{N}(i,k-1)}$ to all $j \in \mathcal{N}(i,1)$;
4 **end**
5 Solve

$$x^{(\mathcal{N}(i,K))} = \begin{array}{ll} \arg\min\limits_{x \in \mathbb{R}^{|\mathcal{N}(i,K)|}} & \sum\limits_{j \in \mathcal{N}(i,K)} f_j(x_j) \\ \text{s.t.} & A_{C_{\mathcal{N}(i,K)}} x = b_{C_{\mathcal{N}(i,K)}} \end{array} \tag{4}$$

$\hat{x}_i = x_i^{(\mathcal{N}(i,K))}$

D. Discussion

It follows directly from the locality analysis in Theorem III.1 that if an accuracy of $\|\hat{x} - x_*\|_\infty \le \varepsilon$ requires

$$K \ge \frac{1}{1-\lambda} \log\left(\frac{C}{\varepsilon}\right)$$

communication rounds. This bound not only determines how to select the number of communication rounds (passed in as a hyperparameter), but provides guidance in determining whether the locality-aware algorithm is suitable for a particular setting. If K is greater than the radius of the network, at least one node has accumulated the entirety of the problem data, and if K is greater than the diameter of the network, every node accumulates and solves the entire problem—in such settings, the locality-aware algorithm may not be suitable. Generally, the locality-aware algorithm offers an advantage in scenarios where the locality parameter, λ, is sufficiently small, and the network diameter is much larger than K.

In contrast to algorithms where estimates of the primal or dual solutions are passed between agents, the message complexity of the proposed algorithm is not constant across iterations—the size of the messages grows at each iteration with the number of agents in each expanding neighborhood. Explicitly, if each local function can be fully represented by B bits, a message representing $\{f_i, A_{C_i,*}, b_{C_i}\}$ requires on the order of $\mathcal{O}(B + 4\max_i |S_i| \times \max_j |C_j|)$ bits, where $\max_i |S_i|$ is the maximum number of agents participating in a constraint, and $\max_j |C_j|$ is the maximum number of constraints any agent participates in. Because $|\mathcal{N}(i,k-1)| \le (\max_i |S_i| \times \max_j |C_j|)^{k-1}$, the maximum message size during the kth communication round is on the order of $\mathcal{O}\left((\max_i |S_i| \times \max_j |C_j|)^k\right)$ bits.

Notably, both the number of communication rounds and the message complexity of the locality-aware algorithm do not directly depend on the number of nodes in the network. In contrast, distributed optimization algorithms that rely on passing estimates of the decision variable requires each node to send messages of size $\mathcal{O}(N)$ at every iteration. Moreover, the number of iterations to convergence of such methods tend to scale with the number of nodes in the network (depending on network topology) [1]. While the message complexity of the locality-aware algorithm grows rapidly between iterations, when A is sparse, $|S_i| \ll N$ and $|C_i| \ll M$. This indicates that the locality-aware algorithm offers a significant advantage in settings where $|S_i|$ and $|C_i|$ remain bounded as N and M are increased, i.e., those where a bounded number of agents participate in constraints, and agents participate in a bounded number of constraints regardless of the size of the network.

A shortcoming of Algorithm 1 is that problem data is explicitly shared between agents. At present, its application is limited to settings where preserving the privacy of individual objective functions and constraint sets is not a concern. However, the scalability of the locality-sensitive algorithm in such settings motivates extending these ideas to design algorithms that exploit locality without explicitly sharing problem data, and we highlight as a promising future direction.

IV. Proofs of Main Results

In this section, we provide proof sketches of the main results summarized in Section III. First, in Section IV-A, we derive the relationship between the true solution to Problem (1) (termed the "global problem") and the solution to the problem obtained by restricting Problem (1) to a subset of the variables and the constraints only involving those variables (termed the "local sub-problem"). Explicitly, we show that the solution of the local sub-problem is consistent with the solution of a perturbed version of the global problem. This then allows us to leverage the sensitivity expression in [12] to derive an expression for the difference between the solution to the local sub-problem and the solution to the global problem (henceforth denoted as the "correction factor").

Second, in Section IV-B, we show that the correction factor derived in Section IV-A yields a numerical structure that reflects the underlying structure of the constraints. Specifically, we show that, while the correction factor will typically be dense, it is possible to find a sparse approximation to the correction factor, where the sparsity pattern of the approximation is a function of the sparsity of the constraints, the desired accuracy, and the conditioning of the global problem. We leverage the guarantees of the Conjugate Residual algorithm to derive, *a priori*, both the sparsity pattern and a bound on the accuracy of the approximation. This approach will allow us to identify which elements of a local solution will be unaffected if a sparse approximation of the correction factor is used. Finally, In Section IV-C, we use the results of the previous subsections to characterize the relationship between the quantity of problem data used, and the error in individual components. This will naturally give rise to the metric of locality λ, which we formally present at the end of the section.

A. Relating local sub-problems to the global problem

In this section, we consider sub-problems generated by restricting Problem (1) to a subset of the primal variables and the constraints only involving those variables. In particular, if $S \subseteq V^{(p)}$ is a subset of the primal variables, we define the *local sub-problem induced by S* as:

$$x^{(S)}(b) = \begin{array}{ll} \arg\min\limits_{x^{(S)} \in \mathbb{R}^{|S|}} & \sum\limits_{i \in S} f_i(x_i^{(S)}) \\ \text{subject to} & A_{C_s,S}x^{(S)} = b_{C_s}. \end{array} \tag{5}$$

Our objective in this section is to relate the value of $x^{(S)}$ to $[x^*(b)]_S$, the components S of the global optimum, allowing us to characterize the error in $x^{(S)}$.

We first show that augmenting the local sub-problem with the remaining variables does not change the solution to the local sub-problem. By computing the optimal unconstrained values for cut variables, we can derive the global constraint vector $\hat{b}$ that induces the same value on S, i.e., $x^{(S)} = \left[x^*(\hat{b})\right]_S$. This equivalence allows us to exploit the sensitivity of optimal points of Problem (1) to perturbations in the constraint vector, b, to derive the correction factor that drives the solution of the local sub-problem to that of the global problem. This interpretation is key for making the connection between the "warm-start" scenario presented in [12] (computing $x^*(b)$ given the solution to $x^*(b+p)$) to the "cold-start" scenario considered in this paper (computing $x^*(b)$ without prior knowledge of other optimal solutions). This allows us to develop a more general theory of locality that fully captures the importance of problem data to individual solution components, as opposed to a theory that only captures response to perturbations.

In the following lemma, we show that if the local-sub-problems are augmented with the remaining variables, the solution on the k-hop neighborhood does not change.

Lemma IV.1 (Augmenting the local sub-problems)**.** Let $x^{(S)}$ be the solution to the local sub-problem induced by S, and

$$\hat{x}^{(S)}(b) = \begin{array}{ll} \arg\min\limits_{x \in \mathbb{R}^N} & \sum\limits_{i=1}^{N} f_i(x_i) \\ \text{subject to} & A_{C_s,S}x = b_{C_s}. \end{array} \tag{6}$$

is the solution to the problem including the entire objective function, but only the constraints of the local sub-problem, then $x^{(S)}(b) = \left[\hat{x}^{(S)}(b)\right]_S$.

Proof. This lemma follows from observing that the variables in $V^{(p)} \setminus S$ are entirely unconstrained, and can be optimized independently from those in S. □

By computing the values that the constraints in $V^{(d)} \setminus C_s$ take on without being enforced, we can derive a constraint vector $\hat{b}$ that induces the same optimal solution as the local sub-problem ("implicit constraints").

Lemma IV.2 (Implicit Constraints)**.** Let $\hat{x}^{(S)}$ be defined as in Lemma IV.1, and $\hat{b} = A\hat{x}^{(S)}$. Then,

$$\hat{x}^{(S)} = \begin{array}{ll} \arg\min\limits_{x \in \mathbb{R}^N} & f(x) \\ \text{subject to} & Ax = \hat{b}. \end{array} \tag{7}$$

Proof sketch. The result follows by showing that the feasible set of Problem (7) is a subset of the feasible set of Problem (6). □

Lemma IV.2 allows us interpret solving the local sub-problem as solving a perturbed version of the global problem where b is replaced by $\hat{b}$. This interpretation allows us to leverage the theory developed by Rebeschini and Tatikonda [12] on the sensitivity of optimal points of Problem (1) to finite perturbations in the constraint vector, b, to relate the solution of the local sub-problem to that of the global problem. The main theorem of [12] is reviewed below.

Theorem IV.3 (Sensitivity of Optimal Points - Theorem 1 of [12])**.** Let $f : \mathbb{R}^N \to \mathbb{R}$ be strongly convex and twice continuously differentiable, and $A \in \mathbb{R}^{M \times N}$ have full row rank. For $b \in \text{Im}(A)$, let $\Sigma(x^*(b)) := \nabla^2 f(x^*(b))^{-1}$. Then $x^*(b)$ is continuously differentiable at all $b \in \mathbb{R}^m$, and

$$\frac{\mathrm{d}x^*(b)}{\mathrm{d}b} = D(b) = \Sigma(x^*(b))A^T(A\Sigma(x^*(b))A^T)^{-1}. \tag{8}$$

The above theorem relates the gradient of the optimal solution, $x^*(b)$, to the constraint matrix and the objective function. Critically, Equation (8) holds globally, allowing us to apply the Fundamental Theorem of Calculus to determine the correction factor necessary to correct for finite perturbations in the constraint vector. Precisely, if we let $\Delta = b - \hat{b}$, the correction factor can be expressed as

$$x^*(\hat{b}+\Delta) - x^*(\hat{b}) = \left(\int_0^1 \Sigma(x_\theta)A^T(A\Sigma(x_\theta)A^T)^{-1}d\theta\right)\Delta,$$

where $x_\theta := x^*(\hat{b}+\theta\Delta)$. Consequently, the error in the local solution is precisely this correction factor.

B. Component-wise Sensitivity

In the previous section, we gave a closed-form expression for the error in the solution of the local sub-problem. In this section, we show how the underlying structure of the optimization problem is reflected in the numerical structure of this error. In particular, we leverage the Conjugate Residuals algorithm [15] to generate a sequence of sparse approximations that converge exponentially to the true correction factor while maintaining sparsity patterns that reflect the underlying graph structure of the optimization problem. We establish that a perturbation in the constraints affects the decision variables "closest" to the constraint the most, while the effect of the perturbation decays with the degrees of separation between a decision variable and the constraint. Moreover, we derive an *a priori* bound of the rate of decay.

In the remainder of this section, we will analyze the instantaneous sensitivity of the optimal point

$$\frac{\mathrm{d}x^*(b)}{\mathrm{d}b}\Delta = D(b)\Delta = \Sigma(x^*(b))A^T(A\Sigma(x^*(b))A^T)^{-1}\Delta.$$

In Section IV-C, when we formally define our metric of locality, the results developed in this section will naturally extend to finite perturbations in the constraint vector. For ease of notation, we let $\Sigma = \Sigma(x^*(b))$.

The instantaneous sensitivity expression will allow us to reason about the structural coupling between components of Problem (1), however, the term $(A\Sigma A^T)^{-1}$ will require careful treatment because the inverse of sparse matrices is typically dense. While the structure of $A\Sigma A^T$ is obfuscated when we take the inverse, it is not lost. The insight that allows us to recover the original structure of the problem in the sensitivity expression is that the Conjugate Residuals algorithm can be leveraged to generate structure-preserving sparse approximations to $\delta := (A\Sigma A^T)^{-1}\Delta$. We now provide a cursory overview of the algorithm and relevant guarantees [15, 6.8][2].

a) Conjugate Residuals: For ease of notation, let $M = A\Sigma A^T$. Conjugate residuals (CR) is an iterative Krylov method for generating solutions to linear systems, $M\delta = \Delta$. The algorithm recursively generates a sequence of iterates

$$\delta^{(k)} \in \mathscr{K}(M,\Delta,k) := \mathrm{span}\{\Delta, M\Delta, M^2\Delta, \ldots, M^{k-1}\Delta\}$$

where each $\delta^{(k)}$ minimizes the norm of the residuals, $\|r_k\| := \left\|\Delta - M\delta^{(k)}\right\|_2$, in the kth Krylov subspace. The guarantees of the algorithm that we will leverage are as follows.

1) Sparsity:
$\delta^{(k)} \in \mathscr{K}(M,\Delta,k) := \mathrm{span}\{\Delta, M\Delta, M^2\Delta, \ldots, M^{k-1}\Delta\}$.
2) Convergence rate:

$$\|r_k\|_2 \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|r_0\|_2 = 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|\Delta\|_2.$$

The first guarantee will allow us to derive the support of each $\delta^{(k)}$, which reflects the underlying structure of the global problem. The second guarantee will allow us to prove the rate with which the effect of a perturbation decays with each degree of separation.

b) Support of the estimates:

Theorem IV.4 (Sparsity Structure of Matrix Powers)**.** For $k \in \mathbb{Z}_+$, neglecting numerical cancellation[3],

$$\mathrm{supp}((A\Sigma A^T)^k) = \{(i,j) \mid d_{G_{\mathrm{con}}}(v_i,v_j) \le k\}.$$

This theorem establishes that the sparsity pattern of a symmetric matrix to the kth power is determined by the k-hop neighbors in the graph representing the sparsity pattern of the original matrix. This allows us the characterize the sparsity pattern of each of the generating vectors of the kth Krylov subspace generated by $A\Sigma A^T$ and Δ.

Corollary IV.4.1 (Sparsity Structure of the Sensitivity Expression)**.** For $k \in \mathbb{Z}_+$ and $i \in [M]$

$$\mathrm{supp}\left(\Sigma(x)A^T\delta^{(k)}e_i\right) \subseteq \mathscr{N}_1^{G_{\mathrm{opt}}}(\mathscr{N}_{k-1}^{G_{\mathrm{con}}}(i)).$$

Informally, $\mathscr{N}_1^{G_{\mathrm{opt}}}(\mathscr{N}_{k-1}^{G_{\mathrm{con}}}(i))$ represents the components of $\Sigma A^T\delta^{(k)}e_i$ that can be deduced to be nonzero based on combinatorial analysis of each of its composing terms. The consequence of Corollary IV.4.1 is that if we take $\Sigma A^T\delta^{(k)}$ as an approximation to $\Sigma A^T(A\Sigma A^T)^{-1}\Delta$, we know which components of the approximation are guaranteed to be zero, i.e., are invariant to locally supported perturbations in the constraint vector. Based on the previous theorem and its corollary, we define a measure of distance between primal variables and dual variables that characterizes the indirect path, through coupling in the constraints, by which a perturbation in the constraint propagates to primal variables,

$$d(v_i^{(p)}, v_j^{(d)}) := \min\{k \mid i \in \mathscr{N}_1^{G_{\mathrm{opt}}}(\mathscr{N}_{k-1}^{G_{\mathrm{con}}}(j))\}.$$

We also define the distance between sets of primal and dual variables as

$$d(I,J) = \min\{d(v_i^{(p)}, v_j^{(d)}) | v_i^{(p)} \in I, v_j^{(d)} \in J\}.$$

c) Component-wise sensitivity: We will now show that the previous result along with the convergence guarantees of CR can be used to infer the component-wise magnitudes of the sensitivity expression. We will ultimately conclude that these magnitudes decay exponentially with rate $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ with the degrees of separation between a component of x, and the support of Δ, where κ is the condition number of $A\Sigma A^T$.

Theorem IV.5 (Decay in Sensitivity)**.** The component-wise magnitudes of the sensitivity expression can be bounded as

$$\|[D(b)\Delta]_S\|_2 \le C\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{d(S,\mathrm{supp}(\Delta))},$$

where $C = \frac{2\|\Delta\|_2}{\sigma_{\min}(A)}$, and $\kappa = \dfrac{\lambda_{\max}(A\Sigma A^T)}{\lambda_{\min}(A\Sigma A^T)}$.

[2] We adapt the results from [15] slightly because $A\Sigma A^T$ is normal.

[3] When characterizing the sparsity pattern of a matrix, "numerical cancellation" refers to entries that are zeroed out due to the values of the matrix entries, and cannot be deduced to be zero from the combinatorial structure of the matrix alone.

Proof sketch. We consider $\{\delta^{(k)}\}$, the sequence of estimates of $(A\Sigma A^T)^{-1}\Delta$ generated via CR, and $\{\Sigma A^T\delta^{(k)}\}$, the corresponding sparse estimates of the sensitivity expression. The convergence guarantees of the CR iterates allow us to bound the error in each $\Sigma A^T\delta^{(k)}$, while their sparsity allows us to deduce the components of $\Sigma A^T\delta^{(k)}$ that are zero. The insight that we leverage is that if $\left\|\Sigma A^T\left((A\Sigma A^T)^{-1}\Delta-\delta^{(k)}\right)\right\|\leq\varepsilon$ and $\left[\Sigma A^T\delta^{(k)}\right]_i=0$, then $|\left[\Sigma A^T\left((A\Sigma A^T)^{-1}\Delta\right]_i\right|\leq\varepsilon$ □

Theorem IV.5 states that components that are "closest" to the perturbation, i.e., those that participate in the constraints, are most sensitive to the perturbation, and the sensitivity of components decay exponentially according to their degree of separation from the perturbation. Moreover, the decay rate can be bounded by $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$. Theorem IV.5 can be readily extended to bound the effect that perturbations in the constraint vector, b, have on individual components of the correction factor.

Corollary IV.5.1 (Decay in Error)**.** If $\lambda\geq\frac{\sqrt{\kappa(x)}-1}{\sqrt{\kappa(x)}+1}$ for all x, then for $C=\frac{2\|\Delta\|}{\sigma_{\min}(A)}$,

$$\left\|\left[x^*(\hat{b}+\Delta)-x^*(\hat{b})\right]_S\right\|_2\leq C\lambda^{d(S,\mathrm{supp}(\Delta))}.$$

Proof sketch. The proof of this theorem proceeds by plugging the bound of Theorem IV.5 into Equation (8). □

Corollary IV.5.1 extends the results of Theorem IV.5 to establish that the magnitude of the correction factor decays with distance to the perturbation. The authors of [12] characterized a similar decay bound for network flow problems, and demonstrated the potential of such a bound in the context of warm-start optimization. This decay bound extends their results to all linearly-constrained convex optimization problems, and improves on our previous results derived from the infinite series expansion of the sensitivity expression [14].

C. Putting it all together

We now have the technical machinery necessary to establish a notion of locality. In this section, we restrict our attention to local sub-problems induced by a k-hop neighborhood around x_i in G_{dec}. To lighten notation, we let $x^{(k)}$ denote the solution to the local sub-problem induced by the k-hop neighborhood around i (denoted by $x_i^{\left(\mathcal{N}_k^{G_{\mathrm{dec}}}(i)\right)}$ in Section IV-A). In this section, we find constants C and λ such that

$$|x_i^{(k)}-x_i^*|\leq C\lambda^k.$$

In other words, we will show that the error in component i decays exponentially according to rate λ with the size of neighborhood generating the local sub-problem. The rate λ naturally characterizes the degree to which local information is sufficient to compute a single component of the global optimum, ultimately, becoming our metric of "locality".

We proceed by leveraging the results of Section IV-A to characterize the error on each of the local sub-problems in terms of the implicit constraints, $\hat{b}^{(k)}$. We will then apply the results derived in Section IV-B to bound the error induced at component x_i. The key difficulty resolved in this section stems from the fact that we want to avoid solving for the implicit constraints (which would require using the entirety of the problem, thus defeating the purpose of locality!)—this is akin to applying Corollary IV.5.1 without knowing Δ.

While we generally cannot control the value of the implicit constraints, $\hat{b}^{(k)}$, the construction of the local sub-problems guarantees that the distance from i to the cut constraints is at least k, i.e., $d(i,\mathrm{supp}(\Delta^{(k)}))\geq k$ where $\Delta^{(k)}:=b-\hat{b}^{(k)}$. Moreover, we know that the "perturbations", $\Delta^{(k)}$, are not arbitrary—they arise from ignoring constraints. These insights provide sufficient knowledge of $\Delta^{(k)}$ to apply Corollary IV.5.1. We are now in a position to prove the main result.

Theorem III.1. Let $x^{(k)}$ be the solution to the optimization problem induced by restricting Problem 1 to k-hop neighborhood around x_i, $\mathcal{N}_k^{(\mathrm{dec})}(i)$, and the constraints only involving those variables. If $\lambda=\sup_x\frac{\sqrt{\kappa(x)}-1}{\sqrt{\kappa(x)}+1}$, where $\kappa(x)$ denotes the condition number of $A\nabla^2f(x)^{-1}A^T$, then

$$|x_i^{(k)}-x_i^*|\leq C\lambda^k \tag{9}$$

for $C=\left(1+\sqrt{\frac{L}{\mu}}\right)\frac{2\sigma_{max}(A)}{\sigma_{min}^2(A)}\left\|b-Ax_{UC}^*\right\|_2$.

Proof sketch. The proof proceeds by first showing that the k-hop sub-problem construction only removes constraints that are at distance k away from component x_i. We then bound the additional constraint violations that can be caused by solving on only a subset of the constraints. For brevity, the proof of this theorem is deferred to the Appendix □

The upshot of this theorem is that if an accuracy of $|x_i^{(k)}-x_i^*|\leq\varepsilon$ is desired, a neighborhood size of

$$K\geq\frac{1}{1-\lambda}\log\left(\frac{C}{\varepsilon}\right)$$

is sufficient. The larger λ is, the larger the neighborhood needed to achieve a desired accuracy, whereas a smaller λ indicates that a smaller neighborhood is sufficient. We note here that the actual number of variables and constraints included in a neighborhood of a fixed size will depend on the problem. For example, if G_{dec} is a path graph, then the number of variables in each neighborhood will scale linearly with k, whereas if G_{dec} is a grid graph, then the number of variables in each neighborhood scales quadratically with k.

The close relationship between λ and the size of sub-problem needed to achieve a desired accuracy justifies it as a metric of the degree to which local information is sufficient to approximate individual components of the global solution. We are now in a position to define our metric of locality.

Definition IV.1 (Locality)**.** For an optimization problem of the form (1) we define the locality of the problem as

$$\lambda(f,A)=\sup_x\frac{\sqrt{\kappa(x)}-1}{\sqrt{\kappa(x)}+1}. \tag{10}$$

We also extend the definition of locality to classes of problems. Explicitly, if it is known that $f\in F$ and $A\in\mathscr{A}$, we define the locality of the class of problems as

$$\lambda(F,\mathscr{A})=\sup_{f\in F,A\in\mathscr{A}}\lambda(f,A). \tag{11}$$

For instance, in network flow problems the class of constraint matrices, $\mathscr{A}$, are those representing flow conservation constraints. The flow conservation constraint at a given node only affects variables for flows departing or arriving at that node; accordingly, the distance metric d corresponds to the shortest-path distance in the network flow graph.

D. Discussion

In this section, we have proposed a metric of locality that captures the amount of information that is required to solve for a single component of a convex optimization problem to a given degree of accuracy. From a practical standpoint, implementing the locality-aware algorithm requires checking the condition number for a given problem instance. In scenarios where the objective function, f, and constraint matrix, A, are fixed, the locality parameter can be computed once, offline, and passed in as a parameter to the network. As an example of such a setting, in Section V we consider an example of economic dispatch, in which we minimize an objective function capturing generation and transmission costs subject to load fulfillment constraints. In such a scenario, the objective function and constraint matrix are fixed while the constraint vector is determined online. Since the objective function and constraint matrix are static, the proposed results can be immediately applied.

In Definition IV.1, we generalize our metric of locality to classes of problems to account problem instances that exhibit variability in the objective and constraint matrix. As an example, in Section V we consider an instance of the power network state estimation problem, in which we maximize the posterior probability of the power flows and voltage angles given noisy measurements of both, subject to the power flow equations. The class of problems encompassing this scenario is defined by objective functions derived from the maximum-a-posteriori estimation formulation, and the constraint matrix encoding the power flow equations. The noisy measurements are modeled in the objective function, so, in contrast with the economic dispatch example, the objective function is stochastic and determined at run-time. We show that the Hessian of the objective function is constant for all possible objective functions of this form. Accordingly, the locality metric can be readily computed in this setting. However, we remark that this is not always be the case, and there is often a practical trade-off between generality of a class of problems and how informative our metric of locality is. For example, if all but one problem in a class exhibit a high degree of locality, the proposed metric would still indicate that the entire class exhibits a low degree of locality—resulting in bounds that are exceedingly conservative for almost all of the problems in that class.

In the case that computing the locality of an entire class of problem is intractable, we suggest a sampling-based approach, where individual problem instances are sampled, and their locality estimated. This motivates a complementary notion of locality in a stochastic sense, where the presented notion of locality is extended from being a worst-case bound to one that captures the distribution of locality parameters in a class of problem. Similarly, we highlight the potential for a class of adaptive algorithms where agents individually estimate local measures of locality based on problem data within their neighborhood (potentially by applying notions of structured and component-wise condition numbers [16]). This not only would alleviate the overhead of computing the global locality parameter, but would remedy the inherent conservatism of worst-case bounds—as demonstrated in Section V, the maximum error of Algorithm 1 across agents can be much worse than the average error.

V. Numerical Experiments

In this section, we validate our theoretical bounds against the true performance of the locality-aware algorithm.

First, we consider an instance of the economic dispatch problem. We compare the true error of the locality-aware algorithm with the theoretical upper-bound on the error, as a function of the number of communication rounds. We observe that when the condition number is low, the performance of the algorithm closely matches the theoretical prediction. We also assess the performance of the projected sub-gradient algorithm and observe that the number of iterations necessary to achieve a high level of accuracy far exceeds the number of communication rounds required for the locally-aware algorithm.

Second, we consider an instance of the rendezvous problem. Intuitively, deciding on a meeting location that is central to all agents is an inherently global problem. This is confirmed by the high locality parameter. Numerically, the rendezvous problem does not exhibit locality that is overlooked by the theory. This confirms that our characterization of locality does not buy us locality when there is none.

A. Economic Dispatch

1) Problem Setting: We consider a setting where generators are positioned in an $N \times M$ grid, and load buses are positioned in the center of each grid cell. Each load bus is only connected to its neighboring generators, which need to supply enough power to satisfy a stochastically generated load $\mathscr{L}(i)$. The costs associated with the problem are a quadratic generation and transmission costs with coefficients $\frac{\alpha}{2}$ and $\frac{\beta}{2}$ respectively. Explicitly, the optimization problem representing this setting is given by

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{\alpha}{2}\sum_i \left(\sum_{j\in\mathscr{N}(i)} x_{i,j}\right)^2 + \frac{\beta}{2}\sum_i \sum_{j\in\mathscr{N}(i)} x_{i,j}^2 \\ \text{subject to} \quad & \sum_{i\in\mathscr{N}(j)} x_{i,j} = \mathscr{L}_j, \forall j. \end{aligned} \tag{12}$$

If $\alpha = 0$, the problem fully decouples and the optimal solution splits each load evenly between its generators. Consequently, this setting allows us to use the parameters α and β to "tune" the locality of the problem and investigate both the tightness of the proposed bounds for varying rates of locality. We note that this example also illustrates the extension of our results to block-separable objectives.

2) Effect of Locality on Convergence: In this example, we fixed the dimension of the global problem to be 20×20, and varied α to be 0.1, 10, and 1000. The condition number for each of these cases was calculated and found to be 1.39, 37.62, and 3611.43 respectively—these correspond to locality parameters of 0.08, 0.72, and 0.97. In each of these cases, we varied the local sub-problem size for each of the agents between 0 and the diameter of the network. Figure 1 plots the maximum error (computed over all the agents) against the size of local sub-problem, as well as the error bound in Theorem III.1 derived from the locality parameter. For well-conditioned problems, the true performance of the algorithm aligns closely with the theoretical prediction, while the theoretical bounds become more conservative as the condition number and the locality parameter increase. Notably, in cases with low locality parameter, the error clear exponential convergence. Whereas, when the locality parameter is higher, the convergence rate of the error appears to increase with the number of communication rounds. This aligns closely with the superlinear convergence behavior sometimes observed with Krylov subspace methods [17].

3) Comparison to other methods: We now evaluate the performance of our algorithm against the standard distributed projected subgradient algorithm [2]. The distributed subgradient algorithm assumes an optimization problem of the form

$$\begin{array}{ll} \underset{x \in \mathbb{R}^N}{\text{minimize}} & \sum_{i=1}^{m} f_i(x) \\ \text{subject to} & x \in \chi_i \end{array} \tag{13}$$

where each $f_i(x)$ and χ_i are only known by agent i, and messages are passed over a fixed communication topology.

Each generator's local objective function encodes its own transmission and generation costs, i.e.,

$$f_i(x) = \frac{\alpha}{2}\left(\sum_{j \in \mathcal{N}(i)} x_{i,j}\right)^2 + \frac{\beta}{2}\sum_{j \in \mathcal{N}(i)} x_{i,j}^2,$$

and each generators' local constraint sets are the load constraints it needs to satisfy. We assume a fixed communication graph where each generator can communicate with other generators that it shares a constraint with—this is exactly the communication graph assumed in Algorithm 1. We use the lazy Metropolis weighting for the consensus step (let L denote the matrix encoding these weights). Every agent maintains and updates a copy of the global variable during each iteration. Let $x_{(i)}^k$ denote the ith agent's copy of the global optimization variable at iteration k. Then the projected subgradient updates are given by

$$x_{(i)}^{k+1} = \Pi_{\chi_i}\left(\sum_j L_{ij} x_{(j)}^k - \frac{\gamma_0}{k^{0.55}} g_{(i)}^k\right),$$

where $\Pi_{\chi_i}(x)$ is the orthogonal projection of the point x on the set χ_i. We simulated the projected subgradient algorithm for varying values of α for 10,000 iterations, and extracted local estimates from each agents' copy of the global decision variable. Figure 2 plots the maximum error across all agents of the projected sub-gradient algorithm against the number of communication rounds. We observe that within 10^4 iterations, none of the estimates have converged to the error achieved by the initial communication round in the locality-aware algorithm despite each agent having access to all of the problem data it would have obtained after the initialization round.

We also note that the convergence of the projected subgradient algorithm is sensitive to the step-size schedule, and that the optimal step size is dependent on the condition number of the problem. Moreover, the best initial step size is not consistent across different problem instances. While it is a weakness that the locality-aware algorithm depends on the condition number, efficient implementation of the projected sub-gradient algorithm also depends on the condition number.

B. Rendezvous

We now consider an instance of rendezvous where 1000 agents, placed randomly in a $[0,1]^2$ grid, must decide on a meeting location the minimizes the sum of their distances to the location. The optimization problem representing this setting is given by

$$\underset{x,y \in \mathbb{R}}{\text{minimize}} \quad \sum_{i=1}^{N} (x - x_i)^2 + (y - y_i)^2. \tag{14}$$

We assume that the communication graph, $G = (V, E)$ between agents is a given by the minimum weight spanning tree of their distances. We rewrite the rendezvous optimization problem in the following form to make it amenable to distributed optimization algorithms,

$$\begin{array}{ll} \underset{\hat{x},\hat{y} \in \mathbb{R}^N}{\text{minimize}} & \sum_{i=1}^{N} (\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 \\ \text{subject to} & \hat{x}_i = \hat{x}_j, \hat{y}_i = \hat{y}_j \quad \forall (i,j) \in E \end{array} \tag{15}$$

This formulation creates local copies of the meeting location coordinates, x and y, and ensures that the neighbors agree on the same meeting location. Because the communication graph is connected, this condition ensures that all agents agree on the same location. As we might expect, deciding on a meeting location that is central to *all* agents is an inherently global problem. This is confirmed by the locality parameter, which was found to be $\lambda = 0.9939$. The true error along with our theoretical bounds are plotted in Figure 3: unlike the example of state-estimation in a power network presented in the appendix, the rendezvous example did not exhibit locality that was overlooked by the theory.

This experiment shows that our characterization of locality does not buy us locality when there is none. Some problems that we might solve with a multi-agent system are inherently global, requiring information from all of the nodes to solve with reasonable accuracy. The purpose of this paper is not to imbue all problems with locality, but rather to develop a metric that can distinguish between the two.

VI. Conclusion

In this paper, we have studied the structure of linearly-constrained strongly-convex optimization problems, showing

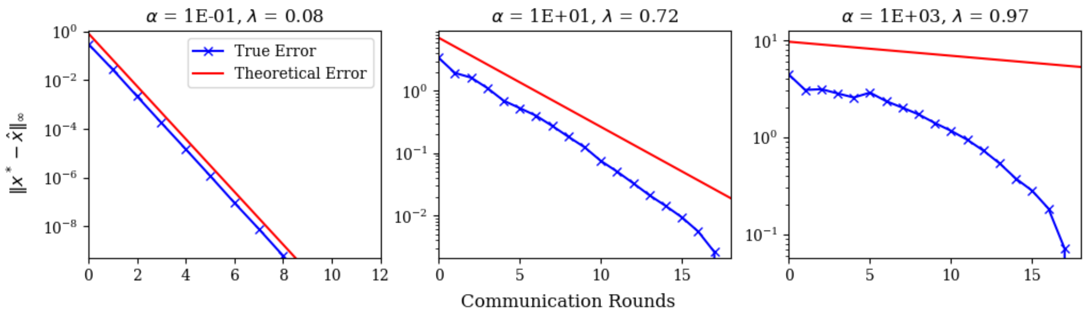

Fig. 1: This figure plots the true accuracy of the locality-aware algorithm (in blue) against the theoretical accuracy (in red) for varying communication rounds. In the well-conditioned case, the proposed theoretical rate is tight. As the conditioning of the problem increases, the theoretical bound becomes more conservative.

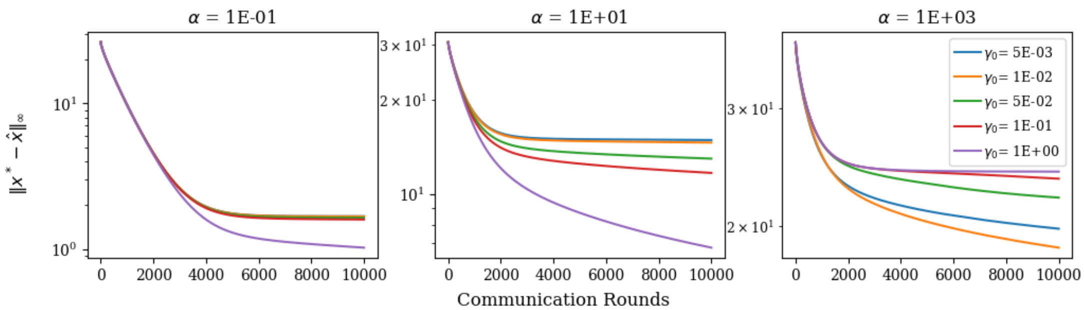

Fig. 2: This figure plots the convergence of the projected sub-gradient algorithm against the number of communication rounds for varying intial step-sizes. The convergence of the algorithm is highly sensitive to the initial step-size.

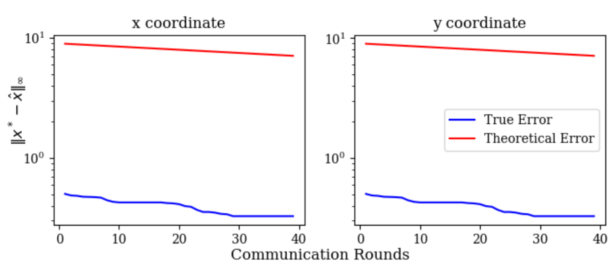

Fig. 3: This figure shows the true accuracy of the locality-aware algorithm (blue) against its theoretical accuracy (red). The locality parameter, $\lambda = 0.9939$, indicates that the error should hardly decay with the number of communication rounds, which aligns with the empirical results observed.

that *all* such problems exhibit locality. Our results leverage Conjugate Residuals to relate the locality of a problem to its conditioning. The rate of locality derived from CR, $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$, is a significant improvement to the $\frac{\kappa-1}{\kappa+1}$ rate derived in previous work via the infinite Neumann expansion. This notion provided a theoretical basis for a rather simple algorithm in which agents individually solve a truncated sub-problem of the global problem. Finally, we demonstrated our algorithm in the context of both economic dispatch and rendezvous.

While the framework of locality appears to be a promising direction for improving the scalability of multi-agent systems, a number of key questions remain open. The first is the issue of determining the locality parameter of a problem—as stated, it is defined as a uniform bound on condition number, which is inherently a global measure. This motivates the question of how to compute the locality of a problem in a distributed fashion. The final question is how we can exploit locality without explicitly sharing problem data between agents.

References

[1] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[2] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[3] D. Jakovetić, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014.

[4] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," in *Proc. IEEE Conf. on Decision and Control*, 2013.

[5] K. I. Tsianos and M. G. Rabbat, "Distributed consensus and optimization under communication delays," in *Allerton Conf. on Communications, Control and Computing*, 2011.

[6] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *Proc. IEEE Conf. on Decision and Control*, 2012.

[7] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[8] D. Varagnolo, F. Zanella, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-raphson consensus for distributed convex optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 994–1009, 2016.

[9] A. Mokhtari, Q. Ling, and A. Ribeiro, "Network newton distributed optimization methods," *IEEE Transactions on Signal Processing*, vol. 65, no. 1, pp. 146–161, 2017.

[10] J. H. Seidman, M. Fazlyab, G. J. Pappas, and V. M. Peciado, "A chebyshev-accelerated primal-dual method for distributed optimization," in *Proc. IEEE Conf. on Decision and Control*, 2018.

[11] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Transactions on Networking*, vol. 52, no. 6, pp. 2508–2539, 2006.

[12] P. Rebeschini and S. Tatikonda, "Locality in network optimization," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 487–500, 2019.

[13] R. Rubinfeld, G. Tamir, S. Vardi, and N. Xie, "Fast local computation algorithms," Massachusetts Inst. of Technology, Tech. Rep., 2011.

[14] R. A. Brown, F. Rossi, K. Solovey, M. T. Wolf, and M. Pavone, "Exploiting locality and structure for distributed optimization in multi-agent systems," in *European Control Conference*, 2020, submitted.

[15] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. SIAM, 2003.

[16] I. Gohberg and I. Koltracht, "Mixed, componentwise, and structured condition numbers," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 3, pp. 688–704, 1993.

[17] B. Beckermann and A. B. J. Kuijlaars, "Superlinear convergence of conjugate gradients," *SIAM Journal on Numerical Analysis*, vol. 39, no. 1, pp. 300–329, 2001.

[18] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel, "Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions," *IEEE Transactions on Signal Processing*, vol. 28, no. 4, pp. 4909–4917, 2013.

[19] C. Josz, S. Fliscounakis, J. Maeght, and P. Panciatici. Ac power flow data in matpower and qcqp format: itesla, rte snapshots, and pegase. Available at https://arxiv.org/abs/1603.01533. Réseau de Transport d'Électricité.

[20] D. Molzahn, F. Dörfler, H. Sandberg, S. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

VII. Appendix

A. Full Proofs of Section IV

Lemma IV.2 (Implicit Constraints)**.** Let $\hat{x}^{(S)}$ be defined as in Lemma IV.1, and $\hat{b} = A\hat{x}^{(S)}$. Then,

$$\hat{x}^{(S)} = \begin{array}{ll} \arg\min & f(x) \\ x \in \mathbb{R}^N & \\ \text{subject to} & Ax = \hat{b}. \end{array} \tag{16}$$

Proof. Assume by contradiction that there exists an optimal solution $\tilde{x}^* \neq \hat{x}^{(S)}$ to Problem (7) with optimal value $f(\tilde{x}^*) < f(\hat{x}^{(S)})$. Note that on $V^{(d)} \setminus C$, the implicit constraints are equal to the true constraints. Precisely, $b_{C_s} = \left[\hat{b}\right]_{C_s}$.

The constraints in Problem (6) are a subset of the constraints in Problem (7). Therefore, the feasible set of Problem (7) is contained in the feasible set of Problem (6). Explicitly,

$$\begin{aligned}\{x \mid Ax = \hat{b}\} &= \{x \mid A_{-C,*}x = \hat{b}_{-C}, A_{C,*}x = \hat{b}_C\} \\ &\subseteq \{x \mid A_{C,*}x = \hat{b}_C\}.\end{aligned}$$

Therefore, if $\tilde{x}^*$ is the optimal solution to Problem (7), it is also a feasible solution for Problem (6). Since $f(\tilde{x}^*) < f(\hat{x}^{(S)})$, $\hat{x}^{(S)}$ is not optimal for Problem (6)—a contradiction. □

Theorem IV.5 (Decay in Sensitivity)**.** The component-wise magnitudes of the sensitivity expression can be bounded as

$$\|[D(b)\Delta]_S\|_2 \leq C \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{d(S,\operatorname{supp}(\Delta))},$$

where $C = \frac{2\|\Delta\|_2}{\sigma_{\min}(A)}$, and $\kappa = \dfrac{\lambda_{\max}(A\Sigma A^T)}{\lambda_{\min}(A\Sigma A^T)}$.

Proof. Let $\delta^{(k)}$ be the kth estimate of $(A\Sigma A^T)^{-1}\Delta$ generated via the Conjugate Residuals algorithm. Corollary IV.4.1 allows us to conclude that $[\Sigma A^T\delta^{(k)}]_S = 0$ if $k \leq d(S, \operatorname{supp}(\Delta))$. It then follows that for all $k \leq d(S, \operatorname{supp}(\Delta))$

$$\begin{aligned}[D(b)\Delta]_S &= [D(b)\Delta - \Sigma A^T\delta^{(k)}]_S \\ &= [\Sigma A^T((A\Sigma A^T)^{-1}\Delta - \delta^{(k)})]_S.\end{aligned}$$

Taking the norm of both sides of the equality, we can bound the sensitivity as

$$\|[D(b)\Delta]_S\|_2 \leq \left\|\Sigma A^T((A\Sigma A^T)^{-1}\Delta - \delta^{(k)})\right\|_2.$$

Notice that the kth residual can be expressed as

$$r_k = A\left(\Sigma A^T\left((A\Sigma A^T)^{-1}\Delta - \delta^{(k)}\right)\right),$$

and convergence of the conjugate residuals algorithms guarantees that

$$\|r_k\|_2 \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|r_0\|_2.$$

Consequently, using the fact that $\sigma_{\min}(A)\|v\| \leq \|Av\|$, we can bound

$$\begin{aligned}\|[D(b)\Delta]_S\|_2 &\leq \left\|\Sigma A^T((A\Sigma A^T)^{-1}\Delta - \delta^{(k)})\right\|_2 \\ &\leq \frac{\|r_k\|_2}{\sigma_{\min}(A)} \leq \frac{2\|\Delta\|_2}{\sigma_{\min}(A)}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k.\end{aligned}$$

Taking $C = \frac{2\|\Delta\|_2}{\sigma_{\min}(A)}$ and $k = d(S, \operatorname{supp}(\Delta))$ concludes the proof. □

Corollary IV.5.1 (Decay in Error)**.** If $\lambda \geq \frac{\sqrt{\kappa(x)}-1}{\sqrt{\kappa(x)}+1}$ for all x, then for $C = \frac{2\|\Delta\|}{\sigma_{\min}(A)}$,

$$\left\|\left[x^*(\hat{b}+\Delta) - x^*(\hat{b})\right]_S\right\| \leq C\lambda^{d(S,\operatorname{supp}(\Delta))}.$$

Proof. Like before, we define $x_\theta := x^*(\hat{b}+\theta\Delta)$, and $b_\theta := \hat{b}+\theta\Delta$. Then,

$$\begin{aligned}&\left\|\left[x^*(\hat{b}+\Delta) - x^*(\hat{b})\right]_S\right\| \\ &= \left\|\left[\int_0^1 \Sigma(x_\theta)A^T(A\Sigma(x_\theta)A^T)^{-1}\Delta d\theta\right]_S\right\| \\ &= \left\|\int_0^1 [D(b)\Delta]_S d\theta\right\| \leq \int_0^1 \|[D(b)\Delta]_S\| d\theta \\ &\leq \int_0^1 \left\|\Sigma(x_\theta)A^T((A\Sigma(x_\theta A^T)^{-1}\Delta - \delta^{(k)})\right\| d\theta \\ &\leq \int_0^1 \frac{2\|\Delta\|}{\sigma_{\min}(A)}\left(\frac{\sqrt{\kappa(x_\theta)}-1}{\sqrt{\kappa(x_\theta)}+1}\right)^k d\theta \leq \frac{2\|\Delta\|}{\sigma_{\min}(A)}\lambda^k.\end{aligned}$$

Taking $C = \frac{2\|\Delta\|}{\sigma_{\min}(A)}$ completes the proof. □

Theorem III.1 (Locality)**.** Let $x^{(k)}$ be the solution to the optimization problem induced by restricting Problem 1 to k-hop neighborhood around x_i, $\mathcal{N}_k^{(\mathrm{dec})}(i)$, and the constraints only involving those variables. If $\lambda = \sup_x \frac{\sqrt{\kappa(x)}-1}{\sqrt{\kappa(x)}+1}$, where $\kappa(x)$ denotes the condition number of $A\nabla^2 f(x)^{-1}A^T$, then

$$|x_i^{(k)} - x_i^*| \leq C\lambda^k \tag{17}$$

for $C = \left(1+\sqrt{\frac{L}{\mu}}\right)\frac{2\sigma_{max}(A)}{\sigma^2_{min}(A)}\left\|b - Ax^*_{UC}\right\|_2$.

Proof. First, we will show that the k-hop local sub-problem can be generated by cutting constrains that are at least distance k from i under the primal-dual distance metric. We will prove this by reasoning about the supports of the appropriate matrix products. The set of primal variables contained in the k-hop neighborhood of x_i can be equivalently characterized as

$$\mathcal{N}_k^{(p)}(i) = \left\{ j \mid \left[(A^TA)^k\right]_{ij} \neq 0 \right\} = \text{supp}([(A^TA)^k]_{i*}).$$

Similarly, the primal-dual distance metric can be defined as

$$\begin{aligned} d(i,c) &= \min\{k \mid c \in \text{supp}\left(\left[A^T(AA^T)^{k-1}\right]_{i*}\right)\} \\ &= \min\{k \mid c \in \text{supp}\left(\left[(A^TA)^{k-1}A^T\right]_{i*}\right)\}. \end{aligned}$$

Because the graph G_{dec} is defined by placing an edge between agents that appear together in the same constraint, if $A_{c,i} \neq 0$ and $A_{c,j} \neq 0$ for some constraint c, then for all $l \in V^{(d)}$,

$$|d(i,l) - d(j,l)| \leq 1.$$

Moreover, to generate the k-hop local sub-problem, a constraint only cut if it contains a variable of distance at least $k+1$. Consequently, all of the primal variables in the cut constraint are at least distance k from i. We can now apply Corollary IV.5.1 to bound the error in component i as

$$|x_i^{(k)} - x_i^*| \leq \frac{2\left\|\Delta^{(k)}\right\|}{\sigma_{\min}(A)}\lambda^k.$$

We will bound the $\Delta^{(k)}$ term by deriving the maximum constraint violation error, $\left\|b - \hat{b}^{(k)}\right\|_\infty$. We do so by noting that the solution to the local sub-problems are consistent with the solution to

$$\begin{aligned} \hat{x}^{\mathcal{N}(i,k)} = \quad & \arg\min_{x \in \mathbb{R}^N} \quad f(x) \\ & \text{subject to} \quad A_{C_{\mathcal{N}(i,K)}}x = b_{C_{\mathcal{N}(i,K)}}. \end{aligned} \tag{18}$$

That is, we use the same set of constraints as agent i's k-hop local sub-problem but include all of the variables in the objective function. Precisely,

$$x^{(\mathcal{N}(i,k))} = \left[\hat{x}^{\mathcal{N}(i,k)}\right]_{\mathcal{N}(i,K)}.$$

Consequently, only variables in $\mathcal{N}(i,k)$ are constrained. We define

$$x^*_{UC} = \arg\min_{x \in \mathbb{R}^N} \quad f(x) \tag{19}$$

to be the solution to the unconstrained problem. Then

$$\left[\hat{x}^{\mathcal{N}(i,K)}\right]_i = \begin{cases} x_i^{(\mathcal{N}(i,k))}, & \text{if } i \in \mathcal{N}(i,k) \\ \left[x^*_{UC}\right]_i, & \text{if } i \notin \mathcal{N}(i,k). \end{cases}$$

The individual components of the implicit constraints can be derived as

$$\left[\hat{b}^{(k)}\right]_i = \begin{cases} b_i, & \text{if } i \in C_{\mathcal{N}(i,k)} \\ \left[Ax^{(\mathcal{N}(i,k))}\right]_i, & \text{if } i \notin C_{\mathcal{N}(i,k)}. \end{cases}$$

It then follows that the component-wise constraint violation is given by

$$\left[b - \hat{b}^{(k)}\right]_i = \begin{cases} 0, & \text{if } i \in C_{\mathcal{N}(i,K)} \\ \left[b - A\hat{x}^{(\mathcal{N}(i,K))}\right]_i, & \text{if } i \notin C_{\mathcal{N}(i,K)} \end{cases}$$

Consequently, the maximum constraint violation is equal to

$$\left\|[b - A\hat{x}^{(\mathcal{N}(i,K))}\right\|_\infty.$$

To obtain a uniform bound, we will show that

$$\left\|b - A\hat{x}^{(\mathcal{N}(i,K))}\right\|_2 \leq \left(1+\sqrt{\frac{L}{\mu}}\right)\frac{\sigma_{max}(A)}{\sigma_{min}(A)}\left\|b - Ax^*_{UC}\right\|_2$$

Because f is L-smooth and μ-strongly convex,

$$\frac{\mu}{2}\left\|x - x_{UC}\right\|_2^2 \leq f(x) - f(x_{UC}) \leq \frac{L}{2}\left\|x - x_{UC}\right\|_2^2$$

Moreover, because $f(\hat{x}) \leq f(x)$,

$$\frac{\mu}{2}\left\|\hat{x} - x_{UC}\right\|_2^2 \leq \frac{L}{2}\left\|x - x_{UC}\right\|_2^2.$$

Then, using the triangle inequality,

$$\begin{aligned} \left\|x - \hat{x}\right\|_2 &\leq \left\|x - x_{UC}\right\|_2 + \left\|x_{UC} - \hat{x}\right\|_2 \\ &\leq \left(1+\sqrt{\frac{L}{\mu}}\right)\left\|x - x_{UC}\right\|_2 \end{aligned}$$

Finally, because $\sigma_{min}(A)\left\|v\right\| \leq \left\|Av\right\| \leq \sigma_{max}(A)\left\|v\right\|$ and $b = Ax$,

$$\left\|b - A\hat{x}\right\|_2 \leq \left(1+\sqrt{\frac{L}{\mu}}\right)\frac{\sigma_{max}(A)}{\sigma_{min}(A)}\left\|b - Ax^*_{UC}\right\|_2.$$

□

B. Additional Experiments

1) Power Network—State Estimation: In this section, we consider a state-estimation problem on the Pan European Grid Advanced Simulation and State Estimation (PEGASE) 9241-bus power-network [18], [19]. From a theoretical standpoint, this problem exhibits a high locality rate, which suggests that a locality-aware algorithm will not be useful in this case. However, empirically we observe that the locally-aware algorithm still manages to find a high-quality solution in fairly few rounds, indicating that our bounds can be overly conservative.

We model the power network by a graph $G(V,E)$. We assume that the network is primarily inductive, the voltage amplitudes are fixed to one, and the voltage angle differences between neighboring nodes are small enough to apply the DC power assumption. Then, the power flow P_{ij} on edge $(i, j) \in E$ satisfies

$$P_{ij} = -b_{ij}(\theta_i - \theta_j).$$

We consider a setting where both the voltage angles, θ, and line power flows, P, are measured according to

$$\theta_i^m = \theta_i + \varepsilon_i, \quad P_{ij}^m = P_{ij} + \varepsilon_{ij}$$

where $\varepsilon_i \sim \mathcal{N}(0,\sigma_i^2)$, and $\varepsilon_{ij} \sim \mathcal{N}(0,\sigma_{ij}^2)$, and the true power flow and voltage angles are estimated. Then, the maximum a posteriori estimation problem is given by

$$\begin{aligned} \underset{\hat{\theta} \in \mathbb{R}^{|V|},\, \hat{P} \in \mathbb{R}^{|E|}}{\text{minimize}} \quad & \sum_{i \in V} \left(\frac{\hat{\theta}_i - \theta_i^m}{\sigma_i} \right)^2 + \sum_{(i,j)\in E} \left(\frac{\hat{P}_{ij} - P_{ij}^m}{\sigma_{ij}} \right)^2 \\ \text{subject to} \quad & \left[\, I \mid B \,\right] \left[\frac{\hat{P}}{\hat{\theta}} \right] = 0 \end{aligned} \tag{20}$$

where I is the identity matrix, and B is the network admittance matrix containing the electrical parameters and topology information [20]. We simulated the locality-aware distributed optimization algorithm (Algorithm 1) for $K = 2,\ldots,20$. The average and maximum errors in both the powerflow and voltage angle estimates are shown in Figure 4 along with their theoretical bounds. We found that the condition number of the problem was 6.37×10^6, resulting in a locality rate of 0.9992. The theoretical bounds, in this case, would suggest that the locality-aware approach is not well-suited to the problem setting. However, numerically, we observe that this bound is overly conservative and the problem instance nevertheless exhibits locality behavior. Additionally, we see that the average error tends to be an order of magnitude less than the maximum error exhibited. Our method of analysis resulted in a uniform worst-case bound, however, this experiment demonstrates that the worst case is a poor representation of the average case. Accordingly, we highlight extending the results of this paper to quantify local measures of locality.

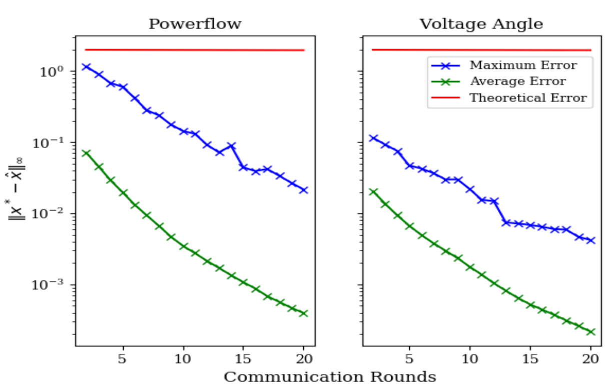

Fig. 4: This figure depicts the local sub-problem size versus average (green), maximum (blue), and theoretical (red) errors in power flow and voltage angle estimates. The theoretical bounds suggest a rate of decay of 0.9992. However, both the maximum and average errors decay much faster, with the average error being consistently an order of magnitude smaller than the theoretical error.