Optimal Querying for Communication-efficient ADMM using Gaussian Process Regression

Aldo Duarte[‡], Truong X. Nghiem[§], and Shuangqing Wei[‡]
[‡] Louisiana State University. [§] Northern Arizona University.

***Abstract*—In distributed optimization schemes consisting of a group of agents connected to a central coordinator, the optimization algorithm often involves the agents solving private local sub-problems and exchanging data frequently with the coordinator to solve the global distributed problem. In those cases, the query-response mechanism usually causes excessive communication costs to the system, leading to the need for communication reduction for scenarios where communication is costly. The integration of Gaussian Processes (GPs) as a learning component to the Alternating Direction Method of Multipliers (ADMM) has been proven successful in learning each agent's local proximal operators, effectively reducing the required communication exchange. A key element for the integration of GP in the ADMM algorithm is the mechanism upon which the coordinator decides when communication with an agent is required. In this paper, we construct a general framework presenting a systematic querying mechanism as an optimization problem that balances reducing the communication cost and minimizing the prediction error. Under this framework, we propose a joint query strategy that takes into account the joint statistics of query and ADMM variables and total communication cost of all the agents in the presence of uncertainty caused by GP regression. Additionally, we derive three different decision mechanisms that simplify the general framework by making the communication decision for each agent individually. We integrate multiple measures to quantify the trade-off between the communication cost reduction and the optimization solution's accuracy/optimality. The proposed methods can achieve significant communication reduction and good optimization solution accuracy for distributed optimization, as demonstrated by extensive simulations of a distributed sharing problem.**

***Index Terms*—distributed optimization, ADMM, proximal operator, communication reduction, Gaussian Process**

I. Introduction

In a distributed optimization scheme that consists of a group of agents connected to a central coordinator, the optimization algorithm often involves the agents solving private local sub-problems and exchanging data frequently with the coordinator. In many of such distributed optimization schemes, the local sub-problems are cast as *proximal minimization problems* [1], which are regularized versions of the original sub-problems, to be solved by the agents in response to queries made by the coordinator. Proximal minimization keeps an agent's local function from being revealed to the coordinator, which is ideal for networks with privacy constraints. Once the coordinator receives the local proximal minimization solutions from the agents, it will use them to calculate new query variables for the agents that keep on driving the agent's objectives to reach the global solution. Applications of distributed optimization include power systems, sensor networks, smart buildings, and smart manufacturing as listed in [2].

An algorithm suited for distributed optimization settings is the Alternating Direction Method of Multipliers (ADMM), first presented in [3]. This algorithm has found great success in distributed optimization because of its simplicity to implement and, due to its decomposing behavior, it is befitting for parallelization. Consequently, ADMM has broad applications in statistical and machine learning problems including the Lasso, sparse logistic regression, basis pursuit, support vector machines, and many others [4]. Examples of usage of ADMM to solve distributed optimization problems include [5], [6], [7], and [8].

The query-response mechanism inherent to distributed optimization schemes (ADMM included) often requires an extensive number of iterations before the algorithm converges to a solution. An extensive number of communications between the coordinator and agents could make the system non-viable in cases where the communication is expensive (e.g. underwater communications for controlling formation of robots [9].) For that reason, reducing the communication expenditure is highly desirable, even critical, for the viability of these distributed optimization schemes in real-life applications. This communication load can be reduced by limiting the number of communication rounds between the coordinator and agents.

Communication reduction in distributed optimization settings has been previously studied. The work in [10] presents a hierarchical distributed optimization algorithm for the predictive control of a smart grid where the communication overhead is reduced by avoiding communication between agents. The works in [11] and [12] proposed general communication-efficient distributed-optimization frameworks for large-scale machine learning applications. In [13], communication-efficient exact and inexact ADMM-based federated learning algorithms are proposed where the aggregation of the nodes to the central coordinator is not performed at each iteration but in defined intervals. The authors of [14] propose a communication-efficient ADMM algorithm to solve a convex consensus optimization problem defined over a decentralized network by using a communication-censoring strategy to alleviate the communication cost.

An alternative approach to communication reduction in distributed optimization via ADMM was proposed in [15] and further developed in [16]. In this approach, the proximal operators of the local agents is predicted so the coordinator can skip some communication rounds. This is achieved using the theory of *the Moreau envelope function* (see, e.g., [17, Chapter 1.G])

and its connections with the proximal operator. This concept was further extended in our previous work [18], where the local proximal operators and their gradients are learned by Gaussian Processes (GPs) with derivative observations. The prediction uncertainties given by the GP models are utilized to decide whether communication between the coordinator and the agents is required. Further communication reduction was achieved in our previous work [19], where the work in [18] was extended to consider Lloyd's and uniform quantization in communications from agents to the coordinator to reduce the payload size of the shared information, thus reducing the overall communication overhead. We further refined our approach in [20], where a linear regression method based on GP was developed to properly account for the impact of the uniform quantization error in learning and predicting with GP.

The mechanism to decide when a communication round should be skipped will affect greatly the desired communication cost reduction and the performance of the ADMM algorithm, therefore developing a systematic approach is critical. Our work in [18] proposed a mechanism to decide whether communication between the coordinator and a particular agent is needed using a heuristic decision method. Such query strategy uses the conditional variance (given by the agent's corresponding GP) and compares it to a threshold that adapts at each algorithmic round depending on the performance of the ADMM algorithm. Though this strategy worked as intended, it was based on an intuitive idea rather than a well-founded systematic querying mechanism. It remains unclear if further communication cost can be reduced by using a more effective querying approach while meeting the constraint in properly solving the underlying ADMM problem, which is the primary question we address in this paper.

Our main contributions are: (1) We propose a systematic querying framework to balance two criteria: reducing the communication overhead and keeping the optimization accuracy. (2) We develop a joint querying method based on the general framework for making joint communication decisions for all agents. (3) We develop three simpler approximate querying strategies, where decisions are made individually to determine which agents are to query. (4) We validate our methods through extensive simulations of a distributed system solving a sharing problem with quadratic cost functions. The results show significant reductions in the total communication expenditure in all test cases compared to the vanilla ADMM approach. Furthermore, all query methods present an acceptable trade-off between communication expenditure reduction and accuracy. However, when comparing each of the method's performance there are clear differences being the joint querying method the one achieving the best results.

Paper Organization: This paper begins with the problem formulation in Section II. The systematic querying framework is presented in Section III. In Section IV, we present our proposed joint query mechanism. The mathematical derivations of our proposed individual query strategies are presented in Section V. A probabilistic comparison between the proposed methods which leads to an expected querying behavior is presented in Section VI. The simulation results are presented in Section VII, and conclusions are made in Section VIII.

II. Problem Formulation

In this manuscript, we solve the sharing problem as considered in [4], [6] with n agents and a central coordinator:

$$\text{minimize} \quad \sum_{i=1}^{n} f_i\left(x_i\right) + h\left(\sum_{i=1}^{n} x_i\right). \tag{1}$$

In this problem, each agent has local decision variables $x_i \in \mathbb{R}^p$ and a convex local cost function $f_i\colon \mathbb{R}^p \mapsto \mathbb{R}$, which are used to minimize the system cost consisting of all the local costs and a convex shared global cost function $h\colon \mathbb{R}^p \mapsto \mathbb{R}$. The cost function f_i can only be known to its corresponding agent. Additionally, the problem in (1) is solved with communication allowed only between the coordinator and agents, but with no exchange between agents.

By introducing copies y_i of x_i, the problem presented in (1) can be solved with the ADMM as shown in Chapter 7.3 of [4] by the iterations

$$\begin{aligned}
x_i^{k+1} &= \underset{x_i\in\mathbb{R}^p}{\arg\min}\left\{f_i(x_i) + (\rho/2)\|x_i - x_i^k - \bar{y}^k + \bar{x}^k + u^k\|^2\right\}\\
\bar{y}^{k+1} &= \underset{\bar{y}\in\mathbb{R}^p}{\arg\min}\left\{h(n\bar{y}) + (n\rho/2)\|\bar{y} - \bar{x}^{k+1} - u^k\|^2\right\}\\
u^{k+1} &= u^k + \bar{x}^{k+1} - \bar{y}^{k+1},
\end{aligned} \tag{2}$$

where $\rho > 0$ is a penalty parameter and $\bar{x}^k = (1/n)\sum_{i=1}^{n} x_i^k$.

At iteration k, each agent will only provide to the coordinator the solution to the following local *proximal minimization problem*

$$\mathbf{prox}_{\frac{1}{\rho}f_i}(z_i^k) = \underset{x_i\in\mathbb{R}^{n_i}}{\arg\min}\left\{f_i(x_i) + \frac{\rho}{2}\|x_i - z_i^k\|^2\right\}, \tag{3}$$

where z_i^k is a value given by the coordinator to the agent i. The x-minimization step consists of the local proximal minimization problem, for every agent i,

$$x_i^{k+1} = \mathbf{prox}_{\frac{1}{\rho}f_i}(\underbrace{x_i^k + \bar{y}^k - \bar{x}^k - u^k}_{z_i^k}).$$

A. STEP-GP Overview

For brevity, we will drop the subscript i and the superscript k in the subsequent equations. The concept of the Moreau envelope of f underlies the STEP (STructural Estimation of Proximal operator) approach in [16] and is defined as

$$f^{\frac{1}{\rho}}(z) = \min_{x\in\mathbb{R}^n}\left\{f(x) + \frac{\rho}{2}\|x - z\|^2\right\}. \tag{4}$$

When f is a convex function, the Moreau envelope $f^{\frac{1}{\rho}}$ is *convex and differentiable with Lipschitz continuous gradient with constant* ρ. Moreover, the unique solution to the proximal minimization $x^{\frac{1}{\rho}}(z) = \mathbf{prox}_{\frac{1}{\rho}f}(z)$ is [21, Proposition 5.1.7]

$$x^{\frac{1}{\rho}}(z) = z - \frac{1}{\rho}\nabla f^{\frac{1}{\rho}}(z). \tag{5}$$

The gradient $\nabla f^{\frac{1}{\rho}}(z)$ is all that is required to reconstruct the optimizer of (3) following from (5). Our previous work [18] improved the STEP method in [16] by learning the local proximal operators with Gaussian Processes (GPs), which are updated online from past query data and used to predict the gradient $\nabla f^{\frac{1}{\rho}}(z)$. This approach is named STructural Estimation of Proximal operator with Gaussian Processes (STEP-GP).

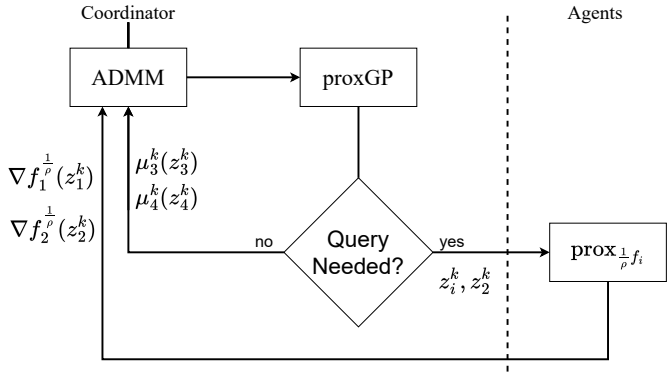

Fig. 1: Flow diagram of the query decision and the query process and response between the coordinator and 4 agents in the proposed approach.

In STEP-GP [18], the Moreau envelope function $f_i^{\frac{1}{\rho}} : \mathbb{R}^{n_i} \mapsto \mathbb{R}$ of each agent is learned by a GP, whose regression accuracy is improved with the incorporation of past queries. In particular, the coordinator will keep a GP model, named proxGP, for every agent. The GP is then used to predict the Moreau Envelope's gradient vector $\nabla f_i^{\frac{1}{\rho}}(z_i^k)$. The GP prediction of the gradient of the Moreau Envelope of agent i has a multivariate Gaussian distribution with conditional mean $\mathbb{E}\left[\nabla f_i^{\frac{1}{\rho}}(z_i^k)\right] = \mu_i^k(z_i^k)$, and conditional covariance matrix $\text{Cov}\left[\nabla f_i^{\frac{1}{\rho}}(z_i^k)\right] = \Sigma_i^k(z_i^k)$. The coordinator decides if a query should be sent to agent i using an uncertainty measurement coming from the conditional covariance matrix. More details of the STEP-GP method can be found in [18].

B. Query-Response Dynamics

In Figure 1, we present one round of the proposed algorithm for a network of 4 agents. The coordinator first calculates the query variables z_i^k for each agent and uses it as input to the agent's corresponding GP regression. The GP regression block named proxGP refers to the GP prediction of $f_i^{\frac{1}{\rho}}(z_i^k)$ and $\nabla f_i^{\frac{1}{\rho}}(z_i^k)$ as presented in [18]. The coordinator has a GP regression running for each agent. In Figure 1 all those GPs are depicted to be in the ProxGP block. Using each of the agent's covariance matrices $\Sigma_i^k(z_i^k)$ given by GP, the coordinator decides which agents require to be queried. In the figure, agents 1 and 2 are set to be queried so the coordinator sends z_1^k and z_2^k to the agents which solve its proximal minimization problems as in (4), depicted in block $\mathbf{prox}_{\frac{1}{\rho}f_i}$. Then, the coordinator receives the corresponding Moreau Envelopes $f_1^{\frac{1}{\rho}}(z_1^k)$, $f_2^{\frac{1}{\rho}}(z_2^k)$ and its gradients $\nabla f_1^{\frac{1}{\rho}}(z_1^k)$, $\nabla f_2^{\frac{1}{\rho}}(z_2^k)$ as a reply from agents 1 and 2. On the other hand, for agents 3 and 4 that are not queried the coordinator uses the corresponding predicted values $\mu_3^k(z_3^k)$ and $\mu_4^k(z_4^k)$ to perform the ADMM updates.

C. ADMM Updates with GP

Following the query-response mechanism presented in Figure 1, the ADMM expression in (2) is modified to include the impact of the GP regression. First, let's define the communication decision variable for agent i at iteration k as

$$\gamma_i^k = \begin{cases} 1, & \text{if agent } i \text{ is queried} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

When $\gamma_i^k = 1$, the query z_i^k is sent to agent i to get the exact value of $\nabla f_i^{\frac{1}{\rho}}(z_i^k)$. On the contrary, when $\gamma_i^k = 0$ we use the predicted value $\mu_i^k(z_i^k)$ given by GP. Next, we can define an expression β_i^k as

$$\beta_i^k = \gamma_i^k \nabla f_i^{\frac{1}{\rho}}(z_i^k) + (1 - \gamma_i^k)\mu_i^k(z_i^k). \tag{7}$$

Consequently, the ADMM expression in (2) can be re-expressed to follow the updates performed at the coordinator's side as:

$$\begin{aligned} x_i^{k+1} &= z_i^k - (1/\rho)\beta_i^k \\ \bar{y}^{k+1} &= \underset{\bar{y}\in\mathbb{R}^p}{\arg\min} \left\{ h(n\bar{y}) + (n\rho/2)\|\bar{y} - \bar{x}^{k+1} - u^k\|^2 \right\} \\ u^{k+1} &= u^k + \bar{x}^{k+1} - \bar{y}^{k+1}. \end{aligned} \tag{8}$$

The work in this manuscript focuses on how we perform the query decision-making, represented in the diamond block in Figure 1.

III. General Querying Decision Framework

The main objective to include GP regression in the ADMM algorithm when solving a distributed optimization problem is to reduce the communication overhead. However, we do not want such a reduction to affect significantly the accuracy of the solution to the problem we try to solve. A key component in the ADMM updates when GP is used, as presented in (8), is the variable β_i^k. This variable becomes either the value of the gradient of the Moreau Envelope $\nabla f_i^{\gamma}(z_i^k)$ or its predicted value. In Equation (8) the set of x^{k+1}, $\bar{y}^k$, and u^{k+1} can be considered as a high dimensional vector trajectory to the global solution. Since GP is being used, the variable β_i^k determines how much we deviate from this trajectory. Furthermore, this variable depends on the communication decision variable γ_i^k as defined in (6) and (7). Therefore, the mechanism to decide each agent's γ_i^k will directly impact the ADMM algorithm's performance. If the coordinator does not have a sound and systematic mechanism to determine when to send queries to the agents, the ADMM algorithm could require excessive iterations to converge or never achieve convergence. Furthermore, it may reach an unsatisfactory solution upon reaching convergence. Thus, the proposed systematic querying method depends on two opposing criteria: 1) Reduce communication overhead, and 2) maintain a good accuracy of the GP regression. Consequently, such query strategy needs to be constrained accordingly to balance those opposing criteria.

Intuitively, we want to solve a constrained optimization of the form

$$\begin{aligned} \text{minimize} \quad & \text{comm}(\gamma^k), \\ \text{subject to} \quad & \gamma_i^k \in \{0,1\}, 1 \le i \le n \\ & \text{uncer}(\gamma^k) \le \psi^k, \end{aligned} \tag{9}$$

where $\text{comm}(\gamma^k)$ is a communication cost function, $\text{uncer}(\gamma^k)$ is an uncertainty cost function, and ψ^k is a given threshold varying at each iteration. The uncertainty cost is compared against this threshold because we want to limit the prediction error at each step. This decision method depends on how we

measure those criteria. We can define a communication cost in several ways, like the number of agents communicating at each iteration or the number of bits exchanged at each communication round. On the other hand, the second criterion could be measured by the uncertainty of the prediction of each agent (given by GP) to control that the communication reduction does not introduce an insurmountable amount of error to the ADMM algorithm. Thus, we can define the query strategy as reducing the communication cost as much as possible constraint limited by the amount of uncertainty introduced by GP.

In general, for given communication and uncertainty cost functions, the optimization problem in (9) has to be solved using combinatorial approaches by seeking the optimal combination of the n binary variables $\{\gamma_i^k, i = 1, \cdots, n\}$. The computation cost could be prohibitive when the number of agents is large. For that reason, we will seek approaches solving the constrained optimization problem in (9) under reasonable communication and uncertainty cost functions without resorting to combinatorial techniques.

IV. Joint Query Method

In this section, we propose a joint query strategy to solve the problem in (9) where the uncertainty cost function is the trace of the joint covariance matrix of the ADMM variables affected by GP regression. The following subsection presents a justification for why the trace is a suitable variable to be used to control the overall prediction error.

A. Justification for Using Trace as a Querying Condition

In this subsection, we derive a norm-based decision strategy about when a query shall be sent to an agent by the coordinator.

Using a general notation, let's define the variable F as a real Gaussian random vector with a predicted mean vector μ and a predicted covariance matrix Σ, where the l^{th} element of μ is μ_l, and the l^{th} element of F is F_l with $l \in [1, \ldots, p]$. Our objective is to determine a proper decision criterion where we want the L2 norm of the discrepancy between the variable F and its predicted mean to be small with high probability. This results in the following confidence sphere:

$$\mathrm{P}\big[||F - \mu||_2 \leq ||\mu||_2 \delta\big] \geq 1 - \xi, \tag{10}$$

where ξ and δ are two small numbers chosen a priori for the purpose of quality control and $||F-\mu||_2 = \sqrt{\sum_{l=1}^{p}(F_l - \mu_l)^2}$. The values of δ and ξ must be small because we want the discrepancy between the actual value and its predicted mean to be small with high probability, so the control variables will determine how tight we allow the discrepancy to be and with how much probability.

We introduce an unitary transformation U, whose column vectors are normalized eigenvectors of the matrix Σ, i.e. $\Sigma = U\Lambda U'$, where Λ is the diagonal matrix whose diagonal entries are eigenvalues of Σ sorted in a descending order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p > 0$. Given $F \sim \mathcal{N}(\mu, \Sigma)$, we define $G = U'(F - \mu)$ which follows $\mathcal{N}(0, \Lambda)$. Moreover, $||G||_2 = ||F - \mu||_2$. Consequently,

$$\mathrm{P}\big[||F - \mu||_2 \leq ||\mu||_2 \delta\big] = \mathrm{P}\big[||G||_2 \leq ||\mu||_2 \delta\big] \geq 1 - \xi. \tag{11}$$

Let us define the variable $Z = \frac{G_l}{\sqrt{\lambda_l}}$ which results in $Z \sim \mathcal{N}(0, 1)$. Then, the probability in (11) can be re-expressed in terms of Z as

$$\mathrm{P}\left[\sum_{l=1}^{p} \lambda_l Z_l^2 \geq ||\mu||_2^2 \delta^2\right] \leq \xi, \tag{12}$$

where instead of requiring a high probability of being inside the confidence sphere, we require the probability of being outside of it to be small.

Defining the variables $Y = \sum_{l=1}^{p} \lambda_l Z_l^2$, which follows a weighted chi-square distribution, and $X = Y - \sum_{l=1}^{p} \lambda_l$ we redefine (12) as

$$\mathrm{P}\left[X + \sum_{l=1}^{p} \lambda_l \geq ||\mu||_2^2 \delta^2\right] \leq \xi. \tag{13}$$

Thus, a sufficient condition to satisfy the condition in (13) is presented in the following proposition.

Proposition 1: A sufficient condition to satisfy the inequality in (10) *is given by*

$$\mathrm{tr}(\Sigma) \leq ||\mu||_2^2 \delta^2 - 2\left(\lambda_1 \ln(1/\xi) + \sqrt{\ln(1/\xi)}\sqrt{\sum_{l=1}^{p} \lambda_l^2}\right). \tag{14}$$

Proof: The proof is presented in Appendix B. □

Proposition 1 shows that we can use the presented bound on the trace of Σ as a decision strategy, which suggests that if the bound is satisfied then we will not communicate with an agent. It should be noted that this bound not only depends on the trace (which is the sum of the eigenvalues) but also on the sum of the squares of the eigenvalues.

B. Proposed Joint Query Method

Following the querying framework presented in (9), we propose using the L_1 norm of $\gamma^k = [\gamma_1^k \ldots \gamma_n^k]$ as the communication cost function, which is a straightforward cost indicating how many agents communicate in the present iteration. On the other hand, in addition to the analysis in the previous subsection, the work in [22] presents a stochastic inexact ADMM approach where the mean-squared error of the inexact ADMM variables with respect to their exact counterpart is bounded. Such bound is presented in their Algorithm 2 where the bounded expectation is the definition of the trace of the error covariance matrix. Extending both analyses to our problem, we have that each agent's uncertainty dynamics are reflected in the updates of the variables of the ADMM algorithm in (8), due to the variables β_i^k and γ_i^k. Thus, we propose to use the trace of the joint uncertainty of the three iterative variables that constitute the ADMM algorithm given by $\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}]$ to derive the global uncertainty cost. The proposed uncertainty cost is $\mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}|\gamma^k])$, where $\mathrm{tr}(.)$ denotes the trace operator. The convexity of both functions is key to the validity of our proposed query strategy.

Thus, we can model our proposed query mechanism similarly to the minimization in (9) as

$$\begin{aligned} \text{minimize} \quad & \|\gamma^k\|_1 \\ \text{subject to} \quad & \gamma_i^k \in \{0, 1\}, 1 \leq i \leq n, \\ & \mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}]|\gamma^k]) \leq \psi^k, \end{aligned} \tag{15}$$

where the threshold ψ^k varies at each iteration and its specifics are presented in Subsection IV-D.

The rationale to use the minimization in (15) is to choose the set of agents that will not be queried such that we minimize the number of communicating agents while ensuring that the joint trace does not exceed the threshold ψ^k, thereby ensuring there is a high probability the uncertainty is within a desired sphere. We next present a solution to the problem in (15) efficiently without resorting to a combinatorial approach by exploiting the convexity and linearity of the considered cost functions and constraints. The idea is that the searching of a set of agents to query starts with the scenario where the communication cost is maximum, and the uncertainty cost is minimum. Then, we calculate the contribution to the joint trace of each agent so the agents that contribute the least to the joint uncertainty will be the first candidates not to be queried in the current round. Instead of considering each possible combination, we analyze the constraint on the joint uncertainty each time the next candidate is set to skip communication until the constraint is met. The proposed joint query method has the following steps at iteration k:

1) Initialize the communication cost to the maximum value by setting all the elements of γ^k to 1.
2) For each agent, its contribution to the trace of the joint uncertainty given by $\mathrm{un}_i = \mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1} | \gamma_i^k = 0, \gamma_{j\neq i}^k = 1])$ is calculated.
3) Sort all un_i in increasing order.
4) In the order from the smallest to the largest un_i, pick all the agents whose total contribution to the uncertainty cost does not exceed the threshold ψ^k and set their γ_i^k to 0, i.e., $\mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1} | \gamma^k]) < \psi^k$.

The proposed strategy does not consider all possible combinations of communicating agents as it would be necessary to combinatorically solve the problem posed in (15). However, our strategy is a good trade-off between the general framework and low computational cost, as evidenced by the numerical results. The next subsections present the specifics of the joint trace $\mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}] | \gamma^k])$ and the threshold ψ^k.

C. Derivation of the Joint Trace Expression

In this subsection, we first present an equivalent expression to the ADMM updates presented in (8) that allow us to see the inherent agent's coupling. This expression is then used to find the specifics of the proposed uncertainty cost $\mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}] | \gamma^k])$. The following proposition uses the notations presented in the problem's definition in Section II.

Proposition 2: The specific form of the ADMM algorithm presented in (8) *has an equivalent expression given by*

$$\begin{aligned} x_i^{k+1} &= z_i^k - (1/\rho)\beta_i^k \\ u^{k+1} &= (1/\rho)\nabla h^{n/\rho}\left(v^k\right) \\ \bar{y}^{k+1} &= \bar{y}^k - 1/(\rho n)\sum_{i=1}^{n}\beta_i^k - u^{k+1}, \end{aligned} \tag{16}$$

where $v^k = n\bar{y}^k - (1/\rho)\sum_{i=1}^n \beta_i^k$ *and* $\nabla h^{n/\rho}()$ *is the gradient of the Moreau Envelope of the function* h.

Proof: The proof is presented in Appendix A. □

The expression in (16) presents the ADMM updates in terms of the gradient of the Moreau Envelope of the functions f and h, and follows the calculations for the ADMM algorithm executed at the coordinator side. More importantly, such an expression also shows that each agent's β_i^k is present in each of the ADMM updates, especially in the $\bar{y}^{k+1}$ and u^{k+1} updates where we have the sum of those variables. The variable β_i^k (depending on γ_i^k as defined in (7)) either comes from the exact value or the predicted value of $\nabla f_i^{\frac{1}{\rho}}(z_i^k)$, so the ADMM updates as presented in (16) can be used to quantify the joint uncertainty of the variables of ADMM.

Due to the linearity of the trace, the proposed uncertainty cost is simplified to $\mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1} | \gamma^k]) = \mathrm{tr}(\mathrm{Cov}[x^{k+1} | \gamma^k]) + \mathrm{tr}(\mathrm{Cov}[\bar{y}^{k+1} | \gamma^k]) + \mathrm{tr}(\mathrm{Cov}[u^{k+1} | \gamma^k])$. Following the expression in (16), the definition of β_i^k in (7), and that only the terms including β_i^k contribute to the uncertainty, we get the expression

$$\begin{aligned} &\mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1} | \gamma^k]) = \\ &\qquad (1/\rho)^2 \sum_{i=1}^{n}(1-\gamma_i^k)\mathrm{tr}\left(\Sigma_i^k(z_i^k)\right) + \\ &\qquad 2(1/\rho)^2 \mathrm{tr}\left(\mathrm{Cov}[\nabla h^{n/\rho}(v^k)]\right) + \\ &\qquad (1/\rho n)^2 \sum_{i=1}^{n}(1-\gamma_i^k)\mathrm{tr}\left(\Sigma_i^k(z_i^k)\right), \end{aligned} \tag{17}$$

The expression in (17) depends on the specifics of the function h.

D. Threshold ψ^k Mechanism

During the execution of the ADMM algorithm, the GP prediction tends to reduce its uncertainty when the ADMM algorithm gets closer to convergence. This is because there is more training data allowing the prediction to be more accurate. For that reason, the threshold to be considered should decrease over the ADMM iterations. We propose a decreasing threshold mechanism that relies on the iteration count and k_0, which is the iteration where the GP regression is used for the first time. At iteration k_0, the initial threshold is

$$\psi^{k_0} = \iota V^{k_0}, \tag{18}$$

where $V_i^{k_0}$ is the uncertainty variable used by the query method (in this case $\mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}] | \gamma^k])$), and ι is a value set manually to set the initial threshold as a percentage of $V_i^{k_0}$. Given a preselected decay rate $\alpha \in (0, 1)$, at a later iteration $k > k_0$, the threshold is calculated as:

$$\psi^k = \psi^{k_0}\alpha^{k-k_0}, \tag{19}$$

where the exponent affecting α increases every iteration making the term α^{k-k_0} smaller every round.

V. Individual Query Methods

In this section, we aim to simplify the query framework presented in Section III. For that reason, we propose different individual query methods to determine when a communication round between the coordinator and agents is necessary. The notation individual query method is used to describe that the coordinator determines if communication with a specific

agent is required by analyzing its uncertainty individually without considering the other agents' uncertainty measures. This strategy reduces considerably the computation complexity of the general method presented in Section III, but ignores the impact of an agent's decision on the overall prediction error introduced to the system. However, by limiting the uncertainty of each agent per iteration we control that the prediction error does not affect the ADMM's algorithm performance greatly. Though this approach is not optimal, its simplicity makes it suitable for applications where the computation cost needs to be as low as possible.

In an individual query method, the decision is performed per agent where such decision is reflected in the agent's corresponding binary decision variable γ_i^k. The general principle of such methods is that for agent i, the controller shall make a decision in favor of not sending a query to this agent if the probability of an estimation error of both the Moreau envelope and its gradients is within an acceptable bound. This estimation error is quantified in different manners. By doing this, we drop the minimization problem presented in (9) and we set each γ_i^k by comparing each agent's estimation error to a threshold. The proposed individual query strategies were not derived arbitrarily but following the mathematical intuition given by a confidence interval analysis to be performed per agent. The specifics of the proposed individual query strategies are presented in the subsequent subsections.

A. Maximum Variance Query Method

Similar to the derivation presented in Section IV-A our goal is to generate a decision rule where the prediction error is small with high probability. For that reason, using the concept of the confidence interval a threshold setting can be derived. When the error of the prediction is below a chosen threshold, there will be no query sent to an agent. As a consequence, we want that the probability of the estimation error being bounded by a small upper bound to be as large as possible.

For the following derivations we employ the general notation used in Section IV-A, where variables F, F_l, μ, μ_l, δ, and ξ were defined, and we add the definition of the vector of variances of F as $s^2 = \text{diag}(\Sigma)$, where the l^{th} element of s^2 is s_l^2. The desired confidence interval is given by

$$\text{P}\left[-|\mu|\delta \leq F - \mu \leq \delta|\mu|\right] \geq 1 - \xi. \tag{20}$$

However, we want to satisfy this condition per dimension of F. Normalizing the interval in (20) we get that the condition per dimension is given by

$$\text{P}\left[\left|\frac{F_l - \mu_l}{s_l}\right| \leq \frac{\delta|\mu_l|}{s_l}, 1 \leq l \leq p\right] \geq 1 - \xi', \tag{21}$$

where $1-\xi' = (1-\xi)^{1/p}$. This variable is introduced to require that the confidence interval shall be satisfied in each dimension. Following the region probability defined in [23], we get an immediate bound of (21) given by:

$$\text{P}\left[\left|\frac{F_l - \mu_l}{s_l}\right| \leq \frac{\delta|\mu_l|}{s_l}, 1 \leq l \leq p\right] \geq \prod_{l=1}^{p} \text{P}\left[\left|\frac{F_l - \mu_l}{s_l}\right| \leq \frac{\delta|\mu_l|}{s_l}\right]. \tag{22}$$

However, instead of analyzing this condition for each of the dimensions of F we can simplify the analysis by requiring that the maximum standard deviation (the maximum element of vector s) satisfy the condition inside the probability in (21) when the bound is minimum. This is attained when

$$\text{P}\left[\frac{|F_l - \mu_l|}{s_l} \leq \frac{\delta|\min_{1\leq l\leq p}(\mu_l)|}{\max_{1\leq l\leq p}(s_l)}, 1 \leq l \leq p\right] \leq 1 - \xi'. \tag{23}$$

The expression in (23) is the same as requiring

$$\max_{1\leq l\leq p}(s_l) \leq \frac{|\min_{1\leq l\leq p}(\mu_l)|\delta}{Q^{-1}(\xi'/2)} = \psi^{(1)}, \tag{24}$$

where $Q^{-1}()$ is the inverse of the Q-function $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-v^2/2} dv$. The right-hand side of the inequality in (24) can be used as the threshold $\psi^{(1)}$ to compare the maximum element of the vector of variances s (s_l). In case $\max_{1\leq l\leq p}(s_l) \leq \psi^{(1)}$, then automatically all the elements of s satisfy the condition.

In the context of the problem defined in Section II, the GP regression gives us at each iteration, and for agent i the predicted mean $\mu_i^k(z_i^k)$ and the conditional covariance matrix $\Sigma_i^k(z_i^k)$. In this scenario, the vector of variances will be defined as $(s_i^k)^2 = \text{diag}(\Sigma_i^k(z_i^k))$. Furthermore, as mentioned in the previous section, each agent's GP prediction uncertainty reduces over the algorithmic rounds. For that reason, the threshold $\psi^{(1)}$ should not be static as it is implied in (24) but should decrease over the ADMM iterations. This requires the control variables ξ and δ to adapt at each iteration which can be problematic considering that the two variables need to be adjusted at each round. Therefore, we do not use the specific threshold $\psi^{(1)}$ defined in (24), but instead, employ a general threshold ψ_i^k per agent which follows the threshold mechanism described in Section IV-D. Finally, under this querying mechanism, the variable γ_i^k is defined as

$$\gamma_i^k = \begin{cases} 0, & \text{if } \max_{1\leq l\leq p}(s_{i[l]}^k) \leq \psi_i^k \\ 1, & \text{otherwise.} \end{cases} \tag{25}$$

B. Maximum Variance and Mean Ratio Query Method

The subsequent proposed strategy expands from the confidence interval analysis presented in Section V-A to build its mathematical intuition. Following the confidence interval defined in (21), to require that each dimension at an agent have a small relative estimation error, we are interested in evaluating the bound in (22). Defining $a^* = \max_{1\leq l\leq p} \frac{s_l}{\mu_l}$, it is then straightforward to show that if

$$\prod_{l=1}^{p} \text{P}\left[\left|\frac{F_l - \mu_l}{s_l}\right| \leq \frac{\delta|\mu_l|}{s_l}\right] \geq \left(\text{P}\left[\left|\frac{F_l - \mu_l}{s_l}\right| \leq \frac{\delta}{a^*}\right]\right)^p \geq 1 - \xi, \tag{26}$$

we always satisfy

$$\text{P}\left[\left|\frac{F_l - \mu_l}{s_l}\right| \leq \frac{\delta|\mu_l|}{s_l}, 1 \leq l \leq p\right] \geq 1 - \xi. \tag{27}$$

Note that under the Gaussian process model, each F_l is Gaussian following $\mathcal{N}(\mu_l, s_l)$. If $\frac{F_l - \mu_l}{s_l}$ follows $\mathcal{N}(0, 1)$, we then obtain a sufficient condition to meet the confidence region requirement stated in (27), namely

$$\max_{1\leq l\leq p} \frac{s_{[l]}}{|\mu_{[l]}|} \leq \frac{\delta}{Q^{-1}(1/2 - 1/2 * (1-\xi)^{1/p})} = \psi^{(2)}, \tag{28}$$

where $Q^{-1}()$ is the inverse of the Q-function $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-v^2}\, dv$. The upper-bound expressed in (28) is imposed not on the maximum element of s but in the maximum ratio of $\frac{s_l}{|\mu_l|}$. The right-hand side of the expression in (28) allows us to define a threshold variable named this time $\psi^{(2)}$.

In the context of our problem defined in Section II, the threshold $\psi^{(2)}$ should decrease over the ADMM algorithmic rounds to keep up with the uncertainty reduction of the GP prediction. Similar to the query method presented in Section V-A, we do not use the specific threshold $\psi^{(2)}$ defined in (28), but instead employ a general threshold ψ_i^k per agent following the mechanism described in Section IV-D. Using the notation of our problem, the variable γ_i^k under this query strategy is defined as

$$\gamma_i^k = \begin{cases} 0, & \text{if } \max\limits_{1\leq l\leq p} \frac{s^k_{i[l]}}{|\mu^k_{i[l]}(z_i^k)|} \leq \psi_i^k \\ 1, & \text{otherwise.} \end{cases} \tag{29}$$

C. Ratio of Maximum Eigenvalue and the Norm of the Mean

In this subsection, we derive a norm-based decision strategy about when a query shall be sent to an agent by the coordinator similar to the one derived in Section IV-D. Our objective is to fulfill the decision criterion presented in (10) given by:

$$\mathrm{P}\big[||F-\mu||_2 \leq ||\mu||_2\delta\big] \geq 1-\xi.$$

Following the same transformation presented in Section IV-D expressed in (11), we seek for an alternative sufficient condition to satisfy the confidence sphere condition in (11). We find an alternative lower bound on this probability by defining $\lambda_1 = \max_{1\leq l\leq p} \lambda_l$ and resorting to the following inequality

$$\sum_{l=1}^{p} \frac{G_l^2}{\lambda_l} \geq \frac{1}{\lambda_1}\sum_{i=1}^{p} |G_l|^2 = \frac{1}{\lambda_1}||G||^2, \tag{30}$$

where $G_l/\sqrt{\lambda_l}$ are i.i.d. standard Gaussian following $\mathcal{N}(0,1)$, which suggests that $\sum_{l=1}^{p} \frac{G_l^2}{\lambda_l}$ follows a chi-square distribution with degree of p, i.e. $\sum_{l=1}^{p} \frac{G_l^2}{\lambda_l} \sim \chi_p^2$. Based on the desired bound in (10) and the inequality in (11), we have a sufficient condition to satisfy (10) given by:

$$\mathrm{P}\big[||G||_2 \leq ||\mu||_2\delta\big] \geq \mathrm{P}\left[\sum_{l=1}^{p} \frac{G_l^2}{\lambda_l} \leq \frac{1}{\lambda_1}||\mu||\delta^2\right] \geq 1-\xi. \tag{31}$$

This expression can be satisfied if λ_1, the maximum eigenvalue of the matrix Σ, satisfies the following condition:

$$\frac{\lambda_1}{||\mu||_2^2} \leq \frac{\delta^2}{\mathcal{F}^{-1}_{\chi^2}(1-\xi)} = \psi^{(3)}, \tag{32}$$

where $\mathcal{F}^{-1}_{\chi^2}(.)$ is the inverse function of the CDF of the chi-square random variable. Thus, if $\frac{\lambda_1}{||\mu||_2^2} \leq \psi^{(3)}$, we ensure that the confidence sphere criterion in (11) is met; thus, there is no need to send a query. It should be noted that different from the approach following a high dimensional confidence region whose sufficient condition is based on the maximum ratio between the standard deviation and its associated absolute mean, as stated in (28), we need to compare the ratio between the maximum eigenvalue and the square of the L-2 norm of the conditional mean to a threshold subject to the chi-square distribution, under the confidence sphere setting.

In the context of the problem defined in Section II, we define the transformation $\Sigma_i^k(z_i^k) = U^k\Lambda_i^k(U^k)'$, where the column vectors of U^k are normalized eigenvectors of the matrix $\Sigma_i^k(z_i^k)$, and Λ_i^k is the diagonal matrix whose diagonal entries are eigenvalues of $\Sigma_i^k(z_i^k)$ sorted in a descending order $\lambda_1^k \geq \lambda_2^k \geq \cdots \geq \lambda_p^k > 0$. Once again, the specific threshold presented in this subsection was not used and was replaced by a general threshold ψ_i^k per agent following the mechanism described in Section IV-D. Finally, we define a query strategy where the variable γ_i^k is defined as

$$\gamma_i^k = \begin{cases} 0, & \text{if } \frac{\lambda_1^k}{||\mu_i^k(z_i^k)||_2^2} \leq \psi_i^k \\ 1, & \text{otherwise.} \end{cases} \tag{33}$$

The query strategies presented in this section are simple strategies with low impact on the overall computational cost, but they ignore the inherent uncertainty dependencies between the agents which will negatively affect the performance of the ADMM algorithm. The following section presents a comparative analysis of the mathematical foundation of each of the proposed methods to have an intuition about what querying behavior to expect for each method.

VI. Method's Probability Comparison

In this small section, we present a comparative analysis of the probabilities used as a foundation of the various querying strategies presented in this work. This analysis allows us to have an idea of the expected querying behavior for each of the methods. For the following derivations, we use the same notation used to derive each of the method's probabilities first defined in Section IV-A.

1) Relationship between Maximum Variance and Maximum Ratio Methods: Comparing the conditions presented in (23) and (26), while acknowledging the bound presented in (22), we can observe that the condition in (23) is more likely to occur. Thus, we have that the relationship between the Maximum Variance and Maximum Ratio between variance and mean methods is given by

$$\left(\mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \leq \frac{\delta}{a^*}\right]\right)^p \leq \mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \leq \frac{\delta|\min_{1\leq l\leq p}(\mu_l)|}{\max_{1\leq l\leq p}(s_l)}, 1\leq l\leq p\right]. \tag{34}$$

This relationship shows that the condition given by the maximum ratio method is more stringent than the one for the maximum variance. For that reason, we anticipate the former to behave more aggressively in terms of the frequency of queries.

2) Relationship between L2 Norm based Methods and an L1 Norm condition: The querying strategies involving the maximum eigenvalue and the trace, presented in Sections V-C and IV-A respectively, are derived starting with the same confidence sphere involving the L2 norm of $F-\mu$. This confidence region is defined in Equation (10). We want to find a relationship between this confidence sphere and a condition involving the L1 norm of $F-\mu$ given by

$$\mathrm{P}\left[||F-\mu||_1 \leq \delta||\mu||_2\right] \geq 1-\xi. \tag{35}$$

We know that for any real vector x, the relationship between L1 and L2 norm is given by $||x||_1 > ||x||_2$. This results in that the

event in (10) is bigger than the event in (35). Furthermore, such an event contains the one in (35). Therefore, if the condition in (10) holds true then automatically the condition in (35) is also true. However, if (35) is true not necessarily (10) is true. For that reason, the condition in (10) is more likely to occur which results in the following relationship

$$\mathrm{P}\left[||F-\mu||_1 \leq \delta||\mu||_2\right] < \mathrm{P}\left[||F-\mu||_2 \leq \delta||\mu||_2\right]. \quad (36)$$

3) Relationship between Maximum Variance Method and an L1 Norm condition: The probability in the condition given in (35) can be expressed as

$$\mathrm{P}\left[\sum_{l=1}^{p}|F_l-\mu_l| \leq \delta||\mu||_2\right]. \quad (37)$$

Since $\delta||\mu||_2$ is a constant we have that the probability of a particular dimension is given by

$$\mathrm{P}\left[|F_l-\mu_l| \leq \frac{1}{p}\delta||\mu||_2,\ 1 \leq l \leq p\right]. \quad (38)$$

The event of the probability in (37) is a bigger event than obtaining the probability given in (38). Therefore, the condition in (37) is more likely to occur giving the following relationship

$$\mathrm{P}\left[|F_l-\mu_l| \leq \frac{1}{p}\delta||\mu||_2,\ 1 \leq l \leq p\right] \leq \mathrm{P}\left[||F-\mu||_1 \leq \delta||\mu||_2\right]. \quad (39)$$

Now, we want to compare the left-hand side of (39) with the probability expression for the Maximum Variance method in (21). Since the variable δ used throughout all the derived probabilities is a variable that can be tuned, we can define a variable $\hat{\delta}$ such that $\frac{1}{p}\hat{\delta}||\mu||_2 = \delta|\min_{1\leq l\leq p}(\mu_l)|$. Dividing by s_l on both sides of the left-hand side of (39), it is straightforward to see the following relationship

$$\mathrm{P}\left[\frac{|F_{[l]}-\mu_{[l]}|}{s_l} \leq \frac{\delta|\min_{1\leq l\leq p}(\mu_l)|}{\max_{1\leq l\leq p}(s_l)}, 1 \leq l \leq p\right] \leq \mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \leq \frac{1}{p}\frac{\hat{\delta}||\mu||_2}{s_l},\ 1 \leq l \leq p\right] \leq \mathrm{P}\left[||F-\mu||_1 \leq \hat{\delta}||\mu||_2\right]. \quad (40)$$

This results in the condition based on the L1 norm of $F-\mu$ being more likely to occur than the condition used in the query method based on the maximum variance.

4) Complete Relationship: Combining the inequalities obtained in (34), (36), and (40) with the definition of $\hat{\delta}$ we get the following relationship

$$\left(\mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \leq \frac{\delta}{a^*}\right]\right)^p \leq \mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \leq \frac{\delta|\min_{1\leq l\leq p}(\mu_l)|}{\max_{1\leq l\leq p}(s_l)}, 1 \leq l \leq p\right] \leq \mathrm{P}\left[||F-\mu||_1 \leq \hat{\delta}||\mu||_2\right] < \mathrm{P}\left[||F-\mu||_2 \leq \hat{\delta}||\mu||_2\right]. \quad (41)$$

The relationship presented in (41) shows how the probabilities used in our proposed decision strategies compare against each other. This relationship allows us to anticipate that the querying dynamics will be more aggressive when using the method based on the maximum ratio of mean and variance, followed by the method based on the maximum variance, and finally, the two methods based on the confidence sphere will present a more relaxed querying dynamics. The following section presents the numerical simulations used to test all the methods presented in this manuscript where we expect to observe a behavior between methods that is congruent with the analysis presented in this section.

VII. Numerical Simulations

In this section, we test the methods proposed in this work by solving a sharing problem where the agent's sub-problems are quadratic. The specifics of the considered sharing problem, the simulation settings, and the results obtained are presented next.

A. Quadratic Problem

1) Problem Definition: We test our methods using a sharing problem motivated by the application presented in [6]. In this work, we did not employ the dynamic behavior of the variables as in [6], but we adapted the problem so the variables are fixed and do not vary at each algorithmic step. The specifics of the considered sharing problem are

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n}[(1/2)x_i^T M_i x_i + w_i^T x_i + c_i] \\ & + (1/2)\sum_{i=1}^{n} y_i^T M_h \sum_{j=1}^{n} y_j + w_h^T \sum_{i=1}^{n} y_i + c_h \\ \text{subject to} \quad & x_i - y_i = 0, \end{aligned} \quad (42)$$

where for $i = 1, \cdots, n$, variables $x_i, y_i \in \mathbb{R}^p$, with $w_i, w_h \in \mathbb{R}^p$, $M_i, M_h \in \mathbb{R}^{p\times p}$ positive definite, and $c_i, c_h \in \mathbb{R}$ being given problem parameters.

2) Calculation of Variables M_i, M_h, w_i, w_h, c_i and c_h: The problem's variables presented (42) are generated following the example given in [6]. First, the variables c_i and c_h will be two randomly generated numbers that are uniformly distributed on [-1,1]. For the case of w_i, we generate for each agent a variable $w_i^{[0]}$ which is a p-dimensional vector with entries randomly generated and uniformly distributed on [-1,1]. Then, the value of w_i is generated for each agent following $w_i = w_i^{[0]} + \eta s_i$, where η is some small positive number, and s_i is a p-dimensional vector for agent i whose entries are randomly generated and uniformly distributed on [-1,1]. The variable w_h is generated following the same procedure as w_i, but it is calculated only once and not for each agent.

On the other hand, to calculate M_i for each agent we first generate a symmetric $p \times p$ matrix $M_i^{[0]} = A * A'$, where the entries of A are randomly generated and uniformly distributed on [-1,1]. Then, we generate $\tilde{M}_i = M_i^{[0]} + \eta S_i$, where $S_i = B * B'$ is a symmetric $p \times p$ matrix with the entries of B being randomly generated and uniformly distributed on [-1,1]. Subsequently, M_i is constructed as

$$M_i = \begin{cases} \tilde{M}_i, & \text{if } \lambda_{min}(\tilde{M}_i) > \epsilon_d \\ \tilde{M}_i + \left(\epsilon_d - \lambda_{min}(\tilde{M}_i)\right) I_p, & \text{otherwise,} \end{cases} \quad (43)$$

where $\lambda_{min}(\tilde{M}_i)$ denotes the smallest eigenvalue of $\tilde{M}_i$ and $\epsilon_d > 0$ is some positive constant. The procedure in (43) is performed to ensure that M_i is positive definite. The variable

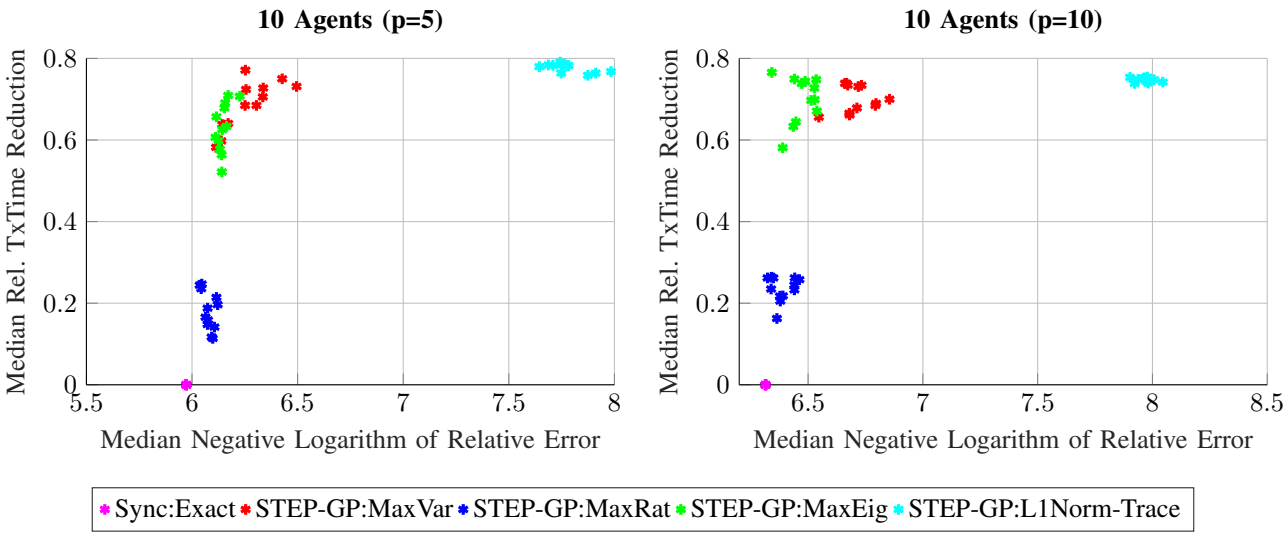

Fig. 2: Performance trade-off between the Relative Transmission Time Reduction and the Negative Logarithm of the Relative Error for 10 Agents with variable's dimension $p = 5$ (left) and $p = 10$ (right). The plots show the 12 best ranked tuple medians of the 100 simulations for different sets of parameters M_i, M_h, w_i, w_h, c_i and c_h, and for different values of α.

M_h is generated following the same procedure as M_i, but it is calculated only once and not for each agent.

3) Solution with ADMM: Following the specifics of the problem in (42) and the ADMM expression in (2), we can derive a closed-form solution for the update of the ADMM's variable x_i^{k+1}. Because the function f_i is convex, the optimal solution of x_i^{k+1} is attained when the gradient of the objective function vanishes. By taking the gradient of the x_i^{k+1}-update and equaling it to zero we obtain

$$x_i^{k+1} = (M_i + \rho I_p)^{-1}(\rho z_i^k - w_i), \tag{44}$$

where I_p is the $p \times p$ identity matrix. The expression in (44) is the closed-form solution of the optimization for the x_i update to be computed at the agent side.

Similarly, we can derive a closed-form solution for the $\bar{y}^{k+1}$ update. Because the function h is also convex quadratic then once again the optimal solution of $\bar{y}^{k+1}$ is attained when the gradient of the objective function vanishes, leading to the expression

$$\bar{y}^{k+1} = (nM_h + \rho I_p)^{-1}(\rho(u^k + \bar{x}^{k+1}) - w_h). \tag{45}$$

Finally, combining the ADMM expression in (2) with the expressions in (44) and (45), we get that the ADMM updates are expressed as

$$\begin{aligned} x_i^{k+1} &= (M_i + \rho I_p)^{-1}(\rho z_i^k - w_i) \\ \bar{y}^{k+1} &= (nM_h + \rho I_p)^{-1}((\rho/n)v^k - w_h) \\ u^{k+1} &= (1/n)(v^k - n\bar{y}^{k+1}), \end{aligned} \tag{46}$$

where $v^k = n\bar{y}^k - (1/\rho)\sum_{i=1}^n \beta_i^k$.

B. Specific form of the joint trace

As presented in Section IV-B, our proposed collective query strategy depends on an uncertainty measurement given by the trace of the joint uncertainty of the ADMM updates. The specific expression of $\text{tr}(\text{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}|\gamma^k])$, following the specific ADMM updates presented in (46), is given by

$$\begin{aligned} \text{tr}(\text{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}|\gamma^k]) = \\ \left((1/\rho)^2 + (1/n\rho)^2\right)\sum_{i=1}^n (1-\gamma_i^k)\text{tr}\left(\Sigma_i^k(z_i^k)\right) + \\ (2/n^2)\sum_{i=1}^n (1-\gamma_i^k)\text{tr}\left(C^T C\Sigma_i^k(z_i^k)\right) - \\ 2(1/n^2\rho)\sum_{i=1}^n (1-\gamma_i^k)\text{tr}\left(C\Sigma_i^k(z_i^k)\right), \end{aligned} \tag{47}$$

where $C = (nM_h + \rho I_p)^{-1}$.

C. Simulation Implementation

The problem in (42) is solved with two different algorithms:

1) *Sync*: this algorithm uses ADMM with proximal operator as in (2), which simplifies to (46) with $\rho = 10$.
2) *STEP-GP*: the algorithm proposed in [18].

For the *STEP-GP* algorithm, different query methods are considered as follows:

- *MaxVar*: The query strategy presented in Section V-A.
- *MaxRat*: The query strategy presented in Section V-B.
- *MaxEig*: The query strategy presented in Section V-C.
- *L1Norm-Trace*: The query strategy presented in Section IV-B.

In our simulations, we consider the following combinations: *Sync*, *STEP-GP:MaxVar*, *STEP-GP:MaxRat*, *STEP-GP:MaxEig*, and *STEP-GP:L1Norm-Trace*. Also, we consider three cases where $n \in \{10, 20, 30\}$.

The simulations were implemented in MATLAB. For comparison purposes, the solution to the minimization problems following (42) are obtained directly using the YALMIP toolbox [24]. For the regression training and inference, we use the GPstuff toolbox [25]. The computation was conducted with high-performance computational resources provided by Louisiana State University (http://www.hpc.lsu.edu).

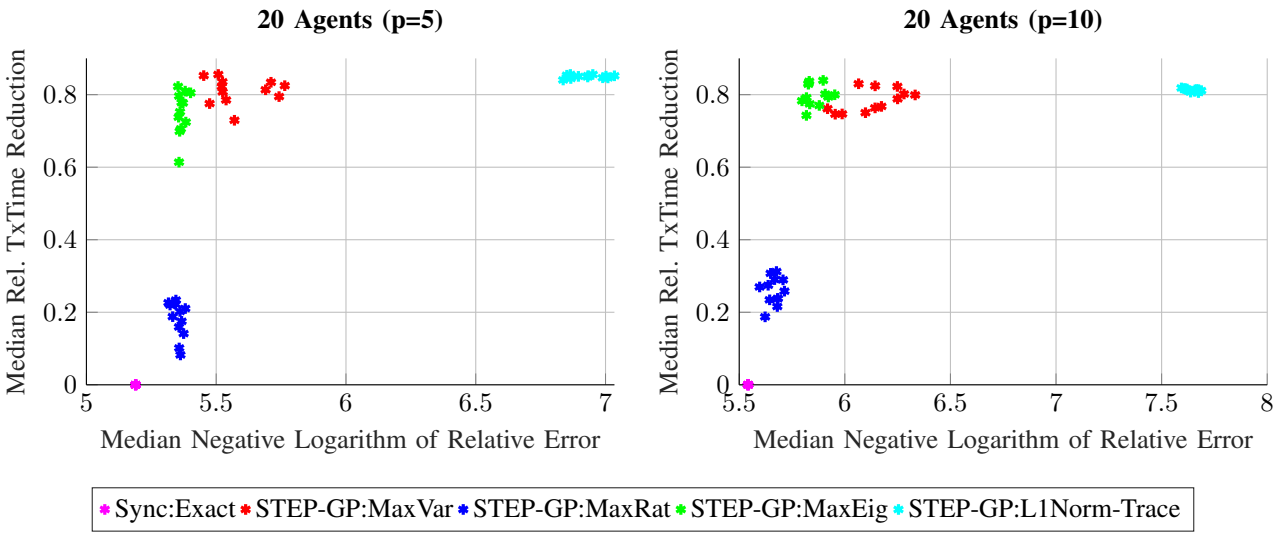

Fig. 3: Performance trade-off between the Relative Transmission Time Reduction and the Negative Logarithm of the Relative Error for 20 Agents with variable's dimension $p = 5$ (left) and $p = 10$ (right). The plots show the 12 best ranked tuple medians of the 100 simulations for different sets of parameters M_i, M_h, w_i, w_h, c_i and c_h, and for different values of α.

D. Metrics and Considerations

1) MAC Metric: We include a simulation component to reflect the channel contention assuming that the coordinator communicates with the agents wirelessly following the IEEE 802.11 specification. We employed the 802.11 CSMA/CA simulator presented in [26], which was implemented in MATLAB. The simulator returns the number of total transmissions, successful transmissions, and an efficiency value defined by $\zeta = st/tt$, where st is the successful transmissions observed and tt the total amount of transmissions performed. The simulation was run offline 1000 times to obtain an average efficiency ζ. At iteration k, the coordinator receives a certain number of simultaneous responses which are expressed in the variable T^k_{simul}. The expected transmission time in one iteration round will be $T^k_{\text{round}} = T^k_{\text{simul}}/\zeta^*$, where ζ^* is the average efficiency in the MAC simulation for the given scenario. The total transmission time is $\text{Tx}_t = \sum_{k=1}^{N} T^k_{\text{round}}$, where N is the number of iterations taken to reach convergence. This metric is not only affected by the total number of communications that were performed but also the number of agents communicating at each iteration, thereby making it a more robust metric to compare the performance of our proposed methods.

2) ADMM Termination Criterion: For our simulations, we use the ADMM termination criterion presented in Section 3.3.1 in [4]. Such criterion presents two conditions comparing the primal and dual of ADMM against two different tolerances. Expressing the primal and dual in terms of the specifics of our problem result in a termination criterion of the form:

$$\|\bar{x}^{k+1} - \bar{y}^{k+1}\|_2 \leq \epsilon^{\text{pri}} \text{ and } \|\rho(\bar{y}^{k+1} - \bar{y}^k)\|_2 \leq \epsilon^{\text{dual}} \quad (48)$$

, where $\epsilon^{\text{pri}} > 0$ and $\epsilon^{\text{dual}} > 0$ are feasibility tolerances for the primal and dual feasibility conditions. These tolerances can be chosen using an absolute and relative criterion, such as

$$\epsilon^{\text{pri}} = \sqrt{p}\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \max(\|\bar{x}^{k+1}\|_2, \|\bar{y}^{k+1}\|_2),$$
$$\epsilon^{\text{dual}} = \sqrt{p}\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \|\bar{y}^{k+1}\|_2, \quad (49)$$

where $\epsilon^{\text{abs}} > 0$ is an absolute tolerance, $\epsilon^{\text{rel}} > 0$ is a relative tolerance, and the factor $\sqrt{p}$ account for the fact that the l_2 norms are in $\mathbb{R}^p$. Both ϵ^{abs} and ϵ^{rel} are manually set at the beginning of the algorithm. The choice of ϵ^{abs} depends on the scale of the typical variable values of the application, while reasonable values for ϵ^{rel} might be 10^{-3} or 10^{-4}, depending on the application.

3) Performance Trade-off: We propose to present the results showing directly the trade-off between the transmission time reduction and the accuracy of the algorithm. Let's define the negative logarithm of the relative error ($NLRE$) expression as

$$NLRE = -\log(|J_{gt} - J_*|/J_{gt}), \quad (50)$$

where J_{gt} is the true optimal value calculated directly with a convex solver, and J_* is the objective value obtained by a particular approach. Also, let's define the relative transmission time reduction (RTx) as

$$RTx = (\text{Tx}_{ADMM} - \text{Tx}_{GP})/\text{Tx}_{ADMM}, \quad (51)$$

where Tx_{ADMM} is the transmission time obtained when running the *Sync:Exact* algorithm, and Tx_{GP} is the transmission time obtained by any of the methods using the *STEP-GP* algorithm. The metric RTx assumes that *Sync:Exact* and the method using *STEP-GP* use the same set of problem's variables.

We present our results in a graph where the vertical axis shows values for RTx and the horizontal axis presents values of $NLRE$. Each point of the graph is a tuple of transmission time reduction and accuracy, and its location indicate how well it performs in terms of the trade-off between these two relative metrics. In particular, the ideal scenario is when $NLRE$ and RTx are both as large as possible. Hence, we want the points to be as close as possible to the right upper corner of the graph.

E. Initial Threshold Tuning

We believe that the variation of the initial threshold has the potential to significantly affect the overall performance of the tested algorithms. For that reason, we propose to fine-tune

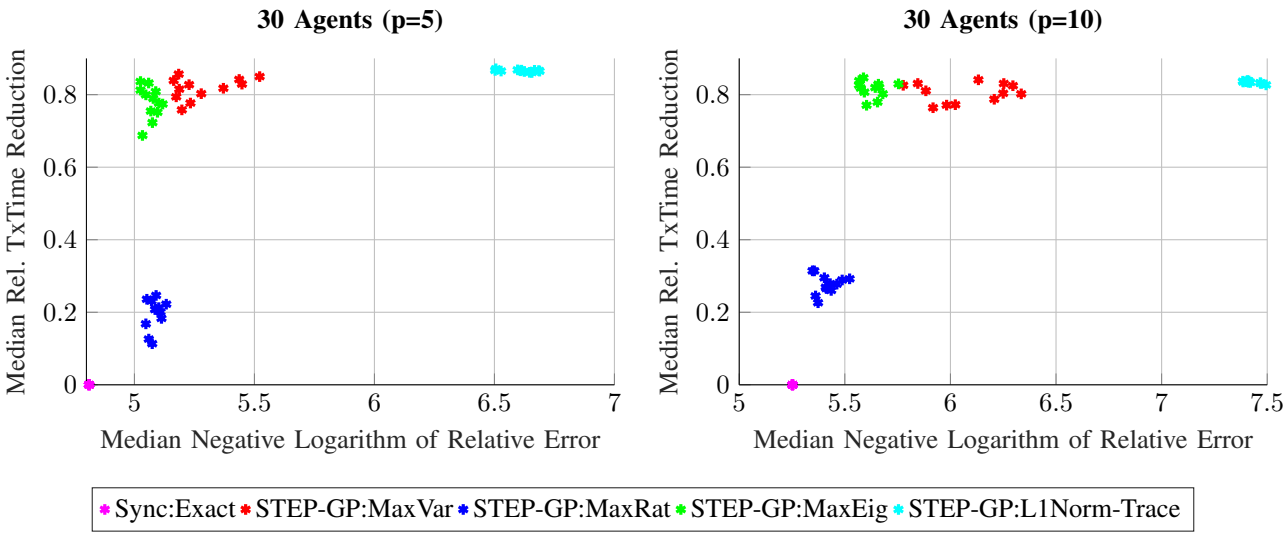

Fig. 4: Performance trade-off between the Relative Transmission Time Reduction and the Negative Logarithm of the Relative Error for 30 Agents with variable's dimension $p = 5$ (left) and $p = 10$ (right). The plots show the 12 best ranked tuple medians of the 100 simulations for different sets of parameters M_i, M_h, w_i, w_h, c_i and c_h, and for different values of α.

the initial threshold for the multiple methods proposed in this work. We consider testing 11 different initial thresholds per case so we can capture the impact of such variation in the proposed methods. The threshold presented in Section IV-D initializes its initial threshold ψ^{k_0} following the expression in (18). Such initialization requires manually setting the variable ι which indicates how proportional concerning V^{k_0} we want ψ^{k_0} to be. For all the different methods tested in this chapter, we considered tunning ψ^{k_0} by considering $\iota = [0.5, 0.6 \ldots, 1.4, 1.5]$.

F. Simulation Results Setting

In this subsection, we present the results for 10, 20, and 30 agents when using the different query strategies proposed in this work with the threshold mechanism described in Section IV-D. We considered different initial threshold values following the description in Section VII-E. Each algorithm for the different methods was run 100 times with different sets of M_i, M_h, w_i, w_h, c_i and c_h generated as in Section VII-A2. In the generated graphs, each point among the same colored cluster represents a tuple of the median values among the 100 simulations of the same method for the $NLRE$ and RTx metrics, as presented in Section VII-D3.

The decaying threshold described in Section IV-D is greatly affected by the selection of the decay rate α. For that reason, we also considered running simulations for different values of α on top of the tuning of the initial threshold. Since we are considering a set of 11 initial thresholds per method, then each tested scenario has 11 points per method and per value of α. The best performance of a given method might occur for a value of α that is not necessarily the same as the rest of the methods. Consequently, we present the results in Figures 2-4 as a ranking of all the median points across all different values of α tested. The ranking is done by setting a tuple as an upper bound with a value of $NLRE$ and RTx that is higher than any of the obtained values in our results. Then, we will calculate the Euclidean distance of all the median points obtained across the different values of α to the upper bound tuple. The 12 median points that attain the lowest distance are included in the graph.

This set of results considered values of $\eta = 0.2$, $\epsilon_d = 1$, $\rho = 10$, $p = 5$, an absolute tolerance value of $\epsilon^{\text{abs}} = 10^{-6}$, a relative tolerance value of $\epsilon^{\text{rel}} = 10^{-5}$, values of $\alpha = [0.95, 0.96, \ldots, 0.99]$, and $x_i^0 = \bar{y}^0 = u^0 = 0$.

G. Simulation Results for 10, 20, and 30 Agents

Figures 2-4 (left) present the $NLRE$ vs RTx graph for 10, 20, and 30 agents of the median of 100 simulations for Sync:Exact and the STEP-GP based algorithms for the different initial thresholds considered, per each of the considered values of α when the variables' dimension is $p = 5$, while Figures 2-4 (right) show the same information but for variables' dimension $p = 10$. The presented results were selected as a consequence of a rank of the best points in terms of the trade-off across all tested values of α. The results in all cases show three main clustering of the presented points. In the lower-left corner appear the points corresponding to STEP-GP:MaxRat in all cases, which presents the worst performance in terms of the trade-off between communication reduction and accuracy. In the upper-left corner, with similar results to each other in all cases, appear STEP-GP:MaxVar and STEP-GP:MaxEig. Those methods present a similar reduction of the transmission time, however STEP-GP:MaxVar presents better relative error values than STEP-GP:MaxEig which is showcased by the points coming from STEP-GP:MaxVar being closer to the ideal case. In the upper-right corner and separated from the other methods appears STEP-GP:L1Norm-Trace with all its points close to each other in all the presented graphs.

On the other hand, the results presented in terms of the relative transmission time reduction in Figures 2-4 correlates

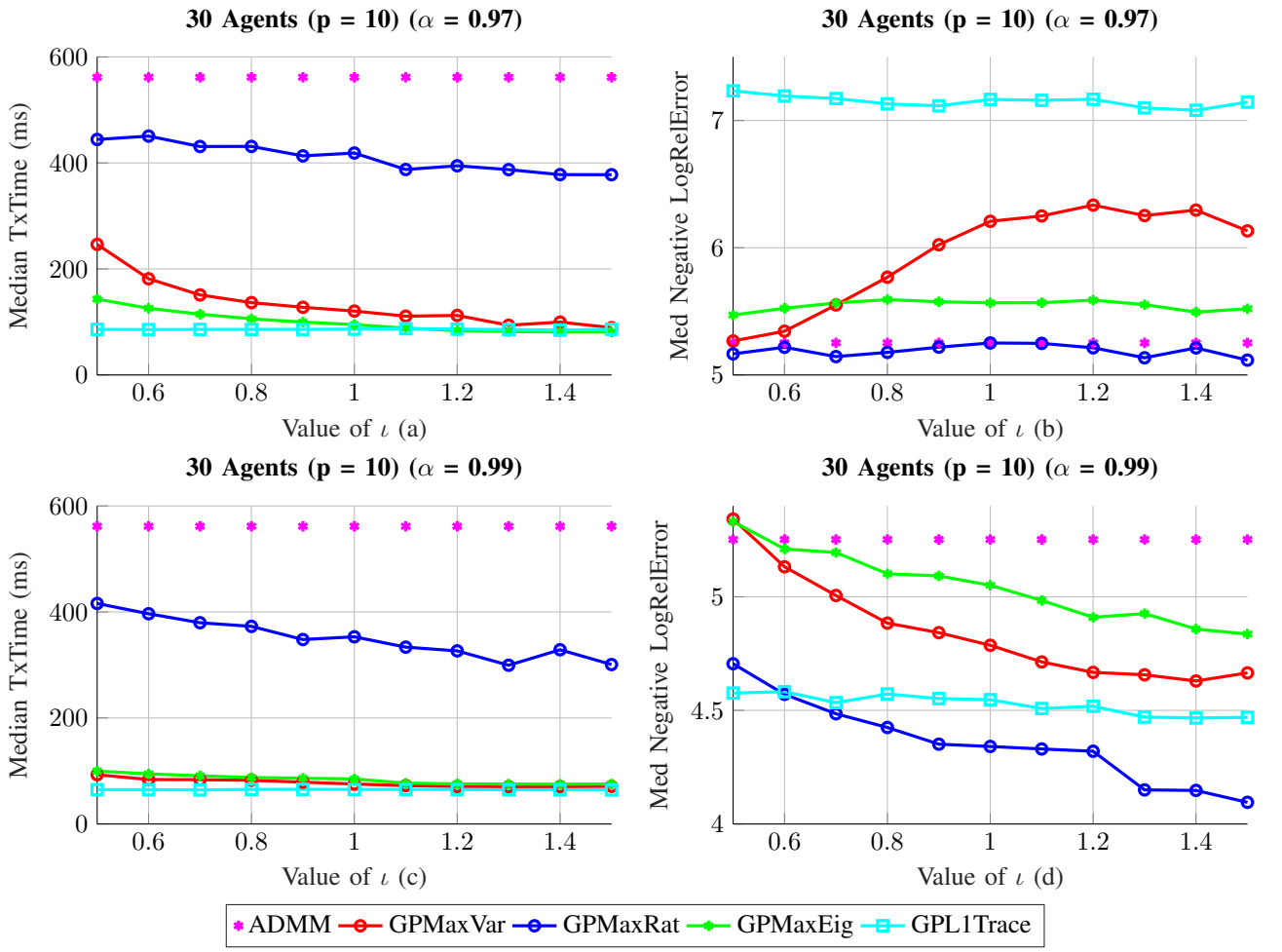

Fig. 5: Results of the median out of 100 simulations for 30 agents and different sets of parameters M_i, M_h, w_i, w_h, c_i, and c_h when those variables dimension is $p = 10$. Graph (a) presents the median values of the Transmission Time for each of the considered values of ι for a decaying rate $\alpha = 0.97$. Graph (b) presents the median values of the Negative Logarithm of the Relative Error for each of the considered values of ι for a decaying rate $\alpha = 0.97$. Graph (c) presents the median values of the Transmission Time for each of the considered values of ι for a decaying rate $\alpha = 0.99$. Graph (d) presents the median values of the Negative Logarithm of the Relative Error for each of the considered values of ι for a decaying rate $\alpha = 0.99$.

with the analysis presented in Section VI. As seen in the graphs, STEP-GP:MaxRat presents the least communication reduction in all cases. The observation of the intermediate results showed that this method asked queries for each agent in around 80% of the total iterations required to reach convergence. Additionally, the two methods based on an L2 norm confidence sphere (STEP-GP:MaxEig and STEP-GP:L1Norm-Trace) present as a cluster a little more relative transmission time reduction than the STEP-GP:MaxVar method. This difference is not significant if we only analyze the relative transmission time reduction metric. However, observing the intermediate results we observed that STEP-GP:MaxEig and STEP-GP:L1Norm-Trace present a lower frequency of queries but require more iterations to converge than STEP-GP:MaxVar. This behavior is more pronounced for the STEP-GP:L1Norm-Trace where the frequency of queries is considerably lower but the increment in number of iterations is also very significant. Thus, the results generated are aligned with the anticipated query behavior.

H. Initial Tuning Impact

The results presented in Figure 5 show the impact of the initial parameters on the overall performance of our proposed methods. The graph presents 4 different plots for the case of 30 agents with variable's dimension $p = 10$. The two upper plots show results of the variation of the median of the transmission time (Graph (a)) and the $NLRE$ metric (Graph (b)) through the different values of ι considered and for a threshold decay rate $\alpha = 0.97$. Graphs (c) and (d) present the same information but for a threshold decay rate $\alpha = 0.99$. In the results presented in the previous subsections, we considered a ranking of the results to present the best results achieved by each method across the tested scenarios. Figure 5 shows that for specific cases, the best performance is not always attained by the method that achieved the best-ranked performance. In all of the cases where the ranked points were presented, STEP-GP:L1Norm-Trace is the one achieving the biggest relative transmission time reduction and lowest relative error. However, when $\alpha = 0.99$

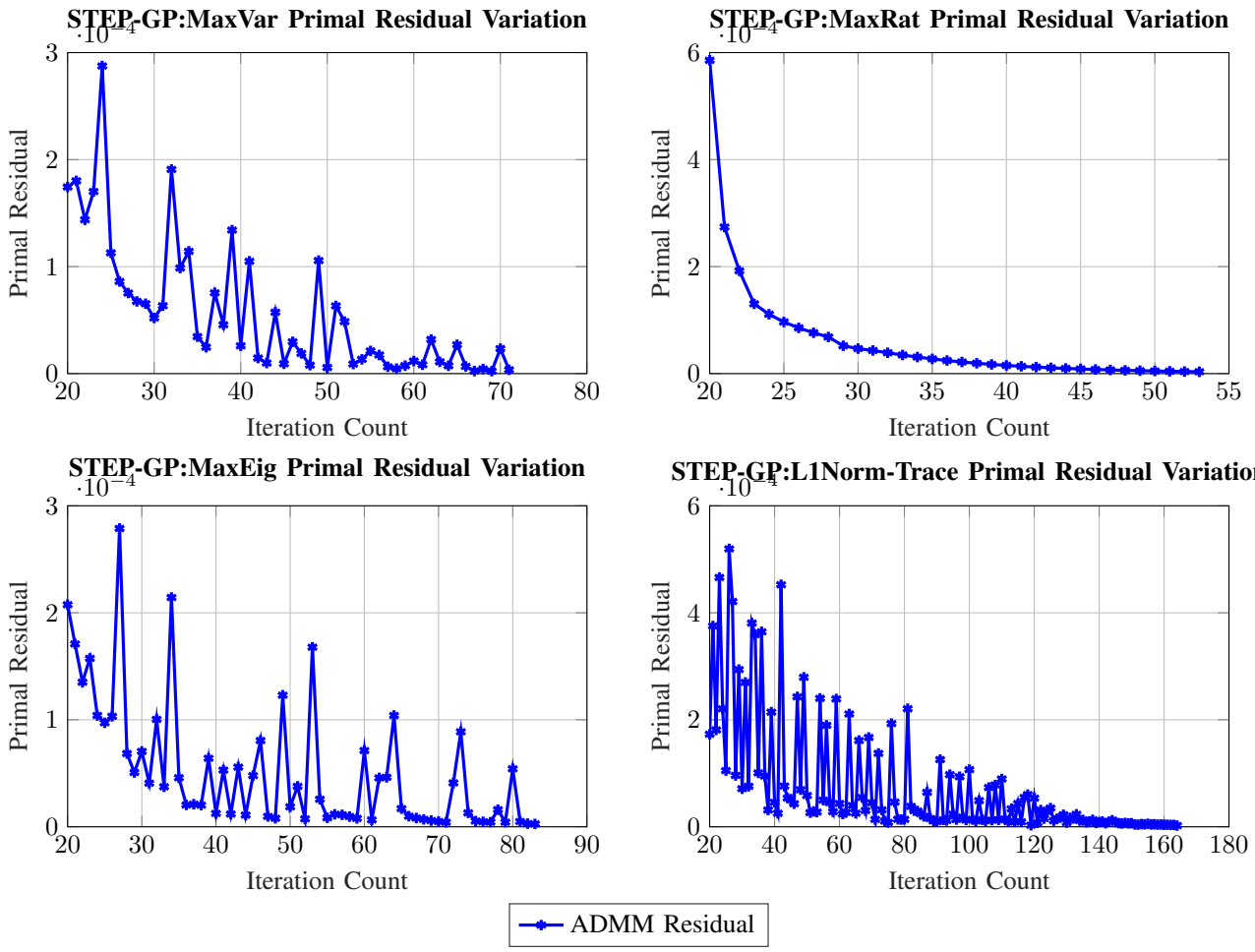

Fig. 6: Variation of the primal residual through the iteration count for all the proposed query methods. The graphs present the test scenario for the same set of parameters M_i, M_h, w_i, w_h, c_i, and c_h of 10 agents with variables' dimension of $p = 10$, an initial threshold given by $\iota = 1$, and decay rate $\alpha = 0.97$ for all cases.

we can see that this method presents worse relative error values than STEP-GP:MaxVar and STEP-GP:MaxEig through all the considered values of ι. This is because under those setting the STEP-GP:L1Norm-Trace method presents a really low query frequency which makes the ADMM algorithm rely heavily upon a GP prediction that is not updated often which, overall, affects the accuracy of our algorithm. Furthermore, if we weigh lower communication cost as more important, overall for all methods choosing $\alpha = 0.99$ would be the best option, where STEP-GP:MaxEig has the best performance in terms of attainable accuracy. However, if we weigh more on the relative accuracy, choosing $\alpha = 0.97$ is the better option, at which the STEP-GP:L1Norm-Trace ends up as the best. This example illustrates the difficulty to present a fair comparison between our proposed algorithms and the challenge of finding the parameters that produce the best results for each one of them.

I. Empirical Convergence

In this subsection, we present an empirical analysis of the convergence of our methods. Figure 6 shows the ADMM primal residual as defined in Section VII-D2 through the iteration count until reaching convergence for all the tested methods. The four graphs present the test scenario for the same set of parameters M_i, M_h, w_i, w_h, c_i, and c_h of 10 agents with variables' dimension of $p = 10$, an initial threshold set by $\iota = 1$, and decay rate $\alpha = 0.97$ for all cases. The presented figures show the decaying behavior of the residual until a significant drop when convergence is achieved. The main difference between methods is the speed of convergence which is defined by the query frequency. The smaller such frequency, the larger the convergence speed. The speed of convergence shown in Figure 6 for each method is aligned with the analysis presented in Sections VI and VII-G. Even though only one case is presented, this trend is observed in all the test scenarios considered in all our experiments presented in the previous subsections. Thus, all the simulations generated (regardless of the test scenario parameters) reached convergence and each query strategy present the same convergence speed behavior.

J. Prediction Error

In this subsection, we present statistics about how the prediction error behaves in our algorithm through all different query methods. Figure 7 presents two graphs showing information on the prediction error of a simulation corresponding to agent 1

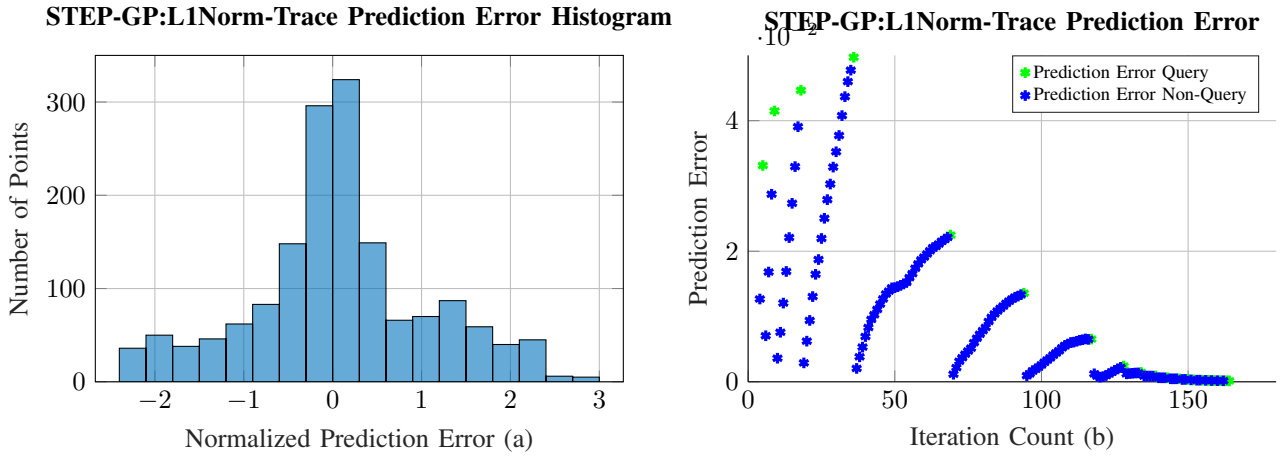

Fig. 7: Prediction Error statistics corresponding to agent 1 under the STEP-GP:L1Norm-Trace query strategy for a specific set of parameters M_i, M_h, w_i, w_h, c_i, and c_h in a system of 10 agents with variables' dimension of $p = 10$, an initial threshold set by $\iota = 1$, and decay rate $\alpha = 0.97$. Graph (a) presents the histogram of the normalized prediction error while graph (b) presents the variation of the L2 norm of the prediction error at each iteration.

under the STEP-GP:L1Norm-Trace query strategy for a specific set of parameters M_i, M_h, w_i, w_h, c_i, and c_h in a system of 10 agents with variables' dimension of $p = 10$, an initial threshold set by $\iota = 1$, and decay rate $\alpha = 0.97$. To generate both graphs we calculated the real values of the Moreau Envelope and its gradient even in iterations where a query was not requested.

In Figure 7 (a) we present the histogram of the normalized prediction error defined as

$$\epsilon^k_{i[NPE]} = \frac{\left| \left[f_i^{\frac{1}{\rho}}(z_i^k); \nabla f_i^{\frac{1}{\rho}}(z_i^k) \right] - \mu_i^k \right|}{s_i^k}.$$

This normalized error results in a vector generated at each iteration for each agent. To construct the presented histogram, we consider each individual component of the vector $\epsilon^k_{i[NE]}$ as a point to be considered in the graph. Following the GP assumptions, we should expect that the discrepancy between Moreau Envelope and its gradient, with the predicted mean follows a Gaussian distribution. However, the histogram in Figure 7 (a) contradicts the prior expectation. This non-normality of the prediction error is also observed in other query strategies throughout different system parameters. Some cases presented histograms showing more discrepancy with respect to the expected Gaussian bell shape than the one presented in Figure 7 (a). This is interesting because these results show that even though the assumed Gaussian distribution of $f_i^{\frac{1}{\rho}}(z_i^k)$ does not hold, the GP is still able to perform a good prediction with acceptable accuracy. Furthermore, this discrepancy with the initial assumption did not prevent any of the tested scenarios to reach convergence.

On the other hand, Figure 7 (b) presents the variation of the L2 norm of the prediction error at each iteration for agent 1. This is defined as

$$\epsilon^k_{i[PE]} = \left\| \left[f_i^{\frac{1}{\rho}}(z_i^k); \nabla f_i^{\frac{1}{\rho}}(z_i^k) \right] - \mu_i^k \right\|_2.$$

This metric generates a single point per iteration, so the presented graph shows the variation of the prediction error over the algorithmic iterations. Figure 7 (b) also makes a differentiation between iterations where a query was made (green points) and iterations where there was no query (blue points). The decaying behavior of the prediction error is clearly seen in the graph with a significant drop closer to convergence. This behavior is desirable because we want our prediction to become more accurate through the algorithmic iterations which is a favorable condition to be confident not only that we reach convergence but that we converge to a good solution. Furthermore, the figure shows a bursting behavior between intervals where we see an increment in the prediction error during the interval where no query was made and an abrupt drop once a query is requested. This behavior of the prediction error is observed for all agents through all the different test scenarios and different query strategies.

K. Query Dynamics

In this subsection, we present information on the distances between the queries z_i^k generated at each iteration compared to the past query points included in the GP training set. Figure 8 (a) presents the measurement of the minimum distance between a new query vector against all the query vectors already in the training set. This distance is defined as

$$d(z_i^k, Z^k) = \inf\{d(z_i^k, z) : z \in Z_i^k\},$$

where Z_i^k is the set containing the queries inside the GP training set for agent i until iteration k and d(.) is the distance function. Since each generated z_i^k is a vector, the distance function considered is $d(z_i^k, Z^k) = ||z_i^k - z||_2$ where $z \in Z_i^k$. Figure 8 (a) presents a differentiation between iterations where a query was made (green points) and iterations where there was no query (blue points). The results show that the distance between the queries throughout the iterations tends to reduce the closer

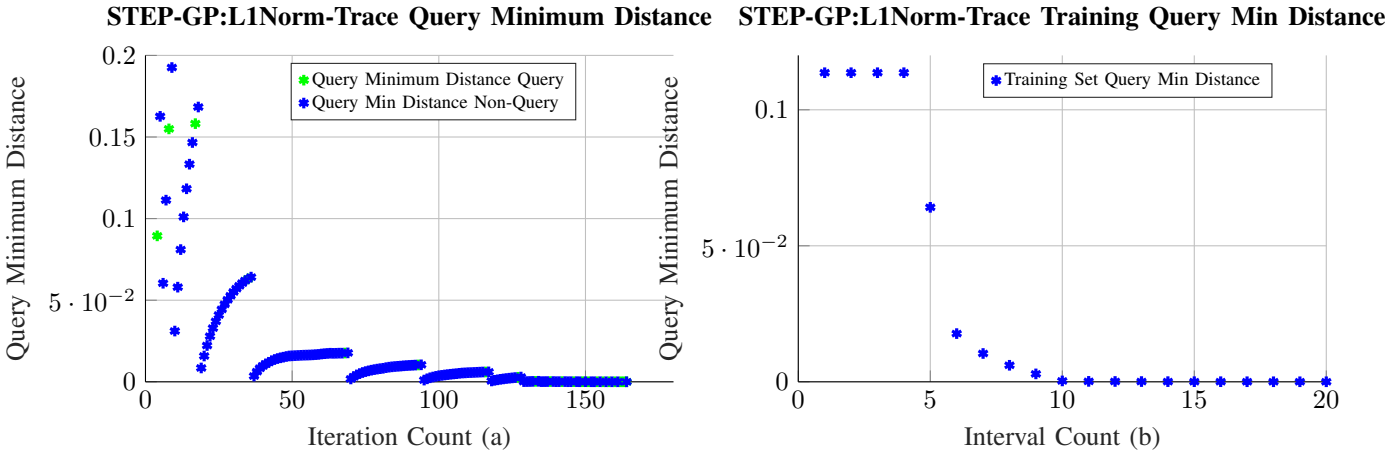

Fig. 8: Distances between generated query points for a specific set of parameters M_i, M_h, w_i, w_h, c_i, and c_h in a system of 10 agents with variables' dimension of $p = 10$, an initial threshold set by $\iota = 1$, and decay rate $\alpha = 0.97$. Graph (a) presents the measurement of the minimum distance between a new query vector against all the query vectors already in the training set. Graph (b) presents the minimum query distances between query points that are already part of the training set only.

we are to convergence. This correlates with the information observed in Figure 7 (b) where the prediction error also reduces the closer we are to convergence. The closer the query points are at the end of the algorithm run the more points are trained in GP around a close vicinity, reducing considerably the uncertainty of the prediction. Also, the behavior of the distance of queries presented in Figure 8 (a) presents a similar bursting behavior as observed for the prediction error in Figure 7 (b).

On the other hand, Figure 8 (b) presents the minimum query distances between the query points already included in the training set. Only when a new point is added to the training set this minimum distance is recalculated. This distance is defined as

$$d(z, x) = \inf\{d(z, x) : z, x \in Z_i^k, z \neq x\},$$

where $d(.)$ once again is defined as $d(z, x) = ||z - x||_2$. The graph in Figure 8 (b) presents a new point when a query is made, so each point presented represents an interval after a period of iterations where no query was made. Similar to the results presented in Figure 8 (b), the distance between query points also decrease closer to convergence. However, in the case where we only compare points that are part of the training set we do not see increasing variations at any point.

L. Overall Remarks

The presented results across different initial parameters showed that the joint query method STEP-GP:L1Norm-Trace is the method that achieved better trade-off performance among all query strategies tested. An observation we made during the simulations is that such a method tends to reduce the required queries considerably, however, it does not require extensive communication rounds to obtain good values for the $NLRE$ metric. When compared to the other tested methods, for similar values of total transmission time the STEP-GP:L1Norm-Trace method usually produces a global ADMM solution closer to the true solution. On the contrary, the STEP-GP:MaxRat method proved to be the one with the worst trade-off performance among all the tested methods. Even though the other individual query strategies presented a similar behavior, it was STEP-GP:MaxVar that presented a better overall trade-off performance compared to STEP-GP:MaxEig. Also, the results obtained were consistent through all the different simulation cases presented. The querying behavior observed during simulations correlates with the previous analysis resulting in an anticipated querying behavior of the proposed methods.

The results presented showed that the more complex querying strategy can achieve the best performance. This outcome agrees with the intuitive idea that the method closer to the general querying framework should achieve better performance. On the other hand, the individual query methods despite their simple strategy were able to maintain an acceptable accuracy while reducing the transmission time considerably. Thus, the individual strategies STEP-GP:MaxVar and STEP-GP:MaxEig are viable options in scenarios where the computation cost needs to be as low as possible.

VIII. Conclusion

Distributed optimization methods such as ADMM usually incur excessive undesired communication overhead. In such context, the use of Gaussian Processes has proven effective in learning the unknown proximal operators of the agents. Therefore, the coordinator can predict the solutions to the local proximal minimization sub-problems, requiring fewer queries to the agents, which leads to a significant reduction in communication. However, the extent of the achievable communication reduction is in part dependent on the mechanism upon which the coordinator decides if communication with the agents is needed. For that reason, this work proposed several query strategies to decide whether the coordinator should send queries

to the agents in a particular iteration when running the *STEP-GP* algorithm based on the notion of the general querying framework. Such an ideal mechanism solves a constrained optimization problem balancing two opposing criteria which are to maximize the communication reduction while minimizing the error of the final solution obtained. Motivated by this systematic method and an alternative expression of the regular ADMM updates showcasing the inherent coupling between agents, we proposed a joint query strategy that consists in minimizing a convex communication cost constrained by the trace of the joint uncertainty of the ADMM variables. On the other hand, to reduce the computational burden added to our algorithm, we proposed different individual query strategies for each agent using an individual uncertainty measure to determine if the prediction is reliable enough to skip a communication round. Numerical simulations of a distributed network solving a sharing problem with quadratic cost functions showed the different performance of the proposed methods in terms of the trade-off between communication reduction and accuracy. In particular, the proposed collective query method achieved a better trade-off performance, when compared with the independent query strategies.

REFERENCES

[1] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[2] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[3] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, 2011.

[5] S. Kumar, R. Jain, and K. Rajawat, "Asynchronous optimization over heterogeneous networks via consensus admm," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 1, pp. 114–129, 2017.

[6] X. Cao and K. J. R. Liu, "Dynamic sharing through the admm," *IEEE Transactions on Automatic Control*, vol. 65, no. 5, pp. 2215–2222, 2020.

[7] Z. Liu, P. Dai, H. Xing, Z. Yu, and W. Zhang, "A distributed algorithm for task offloading in vehicular networks with hybrid fog/cloud computing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2021.

[8] T. Song, D. Li, Q. Jin, and K. Hirasawa, "Sparse proximal reinforcement learning via nested optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4020–4032, 2020.

[9] R. Zhao, M. Miao, J. Lu, Y. Wang, and D. Li, "Formation control of multiple underwater robots based on ADMM distributed model predictive control," *Ocean Engineering*, vol. 257, p. 111585, 8 2022.

[10] P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann, "A distributed optimization algorithm for the predictive control of smart grids," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3898–3911, 2016.

[11] V. Smith, S. Forte, C. Ma, M. Takác, M. I. Jordan, and M. Jaggi, "Cocoa: A general framework for communication-efficient distributed optimization," *arXiv preprint arXiv:1611.02189*, 2016.

[12] C. Ma, J. Konecný, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takác, "Distributed optimization with arbitrary local solvers," *Optimization Methods Software*, vol. 32, no. 4, pp. 813–848, July 2017.

[13] S. Zhou and G. Y. Li, "Communication-Efficient ADMM-based Federated Learning," *arXiv e-prints*, p. arXiv:2110.15318, Oct. 2021.

[14] W. Li, Y. Liu, Z. Tian, and Q. Ling, "Communication-censored linearized admm for decentralized consensus optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 18–34, 2020.

[15] G. Stathopoulos and C. N. Jones, "A coordinator-driven communication reduction scheme for distributed optimization using the projected gradient method," in *Proceedings of the 17th IEEE European Control Conference, ECC 2018, Limassol, Cyprus*, 2018.

[16] G. Stathopoulos and C. Jones, "Communication reduction in distributed optimization via estimation of the proximal operator," *arXiv preprint arXiv:1803.07143*, 03 2018.

[17] R. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Heidelberg, Berlin, New York: Springer Verlag, 1998.

[18] T. X. Nghiem, G. Stathopoulos, and C. Jones, "Learning Proximal Operators with Gaussian Processes," in *Annual Allerton Conference on Communication, Control, and Computing*, Illinois, USA, Oct. 2018.

[19] T. X. Nghiem, A. Duarte, and S. Wei, "Learning-based adaptive quantization for communication-efficient distributed optimization with admm," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 37–41.

[20] A. Duarte, T. X. Nghiem, and S. Wei, "Communication-efficient ADMM using Quantization-aware Gaussian Process Regression," 8 2022.

[21] D. P. Bertsekas, *Convex Optimization Algorithms*. Athena Scientific, 2015.

[22] Y. Xie and U. V. Shanbhag, "Si-admm: A stochastic inexact admm framework for resolving structured stochastic convex programs," in *2016 Winter Simulation Conference (WSC)*, 2016, pp. 714–725.

[23] H. Nagao and M. Srivastava, "Fixed width confidence region for the mean of a multivariate normal distribution," *Journal of Multivariate Analysis*, vol. 81, pp. 259–273, 05 2002.

[24] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. of the CACSD Conference*, Taipei, Taiwan, 2004.

[25] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari, "GPstuff: Bayesian modeling with gaussian processes," *Journal of Machine Learning Research*, vol. 14, pp. 1175–1179, 2013.

[26] N. A. NAGENDRA. (2013) Ieee 802.11 mac protocol. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/44110-ieee-802-11-mac-protocol

[27] A. R. Zhang and Y. Zhou, "On the non-asymptotic and sharp lower tail bounds of random variables," *Stat*, vol. 9, 2018.

[28] B. Laurent and P. Massart, "Adaptive estimation of a quadratic functional by model selection," *The Annals of Statistics*, vol. 28, no. 5, pp. 1302–1338, 2000. [Online]. Available: http://www.jstor.org/stable/2674095

[29] L. Birgé and P. Massart, "Minimum contrast estimators on sieves: Exponential bounds and rates of convergence," *Bernoulli*, vol. 4, no. 3, pp. 329–375, 1998. [Online]. Available: http://www.jstor.org/stable/3318720

APPENDIX
APPENDIX A: PROOF OF PROPOSITION 2

Combining the definition of $z_i^k = x_i^k + \bar{y}^k - \bar{x}^k - u^k$ and the expression for x_i^{k+1} defined in (5), we can express the update of $\bar{y}$ in (8) as

$$\bar{y}^{k+1} = (1/n)\underset{\hat{y}\in\mathbb{R}^p}{\arg\min}\left\{h(\hat{y}) + (\rho/2n)\|\hat{y} - n(\bar{x}^{k+1} + u^k)\|^2\right\},$$

where $\hat{y} = n\bar{y}$. Then, we can express $\bar{y}^{k+1}$ in terms of its proximal operator $\bar{y}^{k+1} = (1/n)\,\mathbf{prox}_{(n/\rho)h}[n(\bar{x}^{k+1} + u^k)]$, which can be expressed in terms of the gradient of the Moreau Envelope of h, as in (5), leading to

$$\bar{y}^{k+1} = (\bar{x}^{k+1} + u^k) - (1/\rho)\nabla h^{n/\rho}\left(n(\bar{x}^{k+1} + u^k)\right). \quad \text{(A.1)}$$

Now, expressing the u-update presented in (2) in terms of (A.1) gives

$$u^{k+1} = (1/\rho)\nabla h^{n/\rho}\left(n(\bar{x}^{k+1} + u^k)\right). \quad \text{(A.2)}$$

Next, we can express (A.1) in terms of z_i^k as

$$\bar{y}^{k+1} = (1/n)\sum_{i=1}^{n}[z_i^k - (1/\rho)\nabla f_i^{1/\rho}(z_i^k)] + u^k - (1/\rho)\nabla h^{n/\rho}\left(n(\bar{x}^{k+1} + u^k)\right), \quad \text{(A.3)}$$

and by inserting the definition of z_i^k we get

$$\bar{y}^{k+1} = \bar{y}^k - 1/(\rho n)\sum_{i=1}^{n}\nabla f_i^{1/\rho}(z_i^k) - (1/\rho)\nabla h^{n/\rho}(n(\bar{x}^{k+1} + u^k)). \quad \text{(A.4)}$$

Taking the average of the definition of z_i^k we get $\bar{z}^k = \bar{y}^k - u^k$, and by inserting it in the average of the x_i-updates given by $\bar{x}^k = \bar{z}^k - 1/(\rho n)\sum_{i=1}^n \nabla f_i^{1/\rho}(z_i^k)$ we get the equality

$$\bar{y}^k - 1/(\rho n)\sum_{i=1}^{n} \nabla f_i^{1/\rho}(z_i^k) = \bar{x}^{k+1} + u^k. \tag{A.5}$$

Thus, combining (A.4) and (A.5) we obtain that

$$\bar{y}^{k+1} = \bar{y}^k - 1/(\rho n)\sum_{i=1}^{n} \nabla f_i^{1/\rho}(z_i^k) - (1/\rho)\nabla h^{n/\rho}\left(n\bar{y}^k - (1/\rho)\sum_{i=1}^{n} \nabla f_i^{1/\rho}(z_i^k)\right), \tag{A.6}$$

and the u-update combining (A.2) with (A.5) is expressed as

$$u^{k+1} = (1/\rho)\nabla h^{n/\rho}\left(n\bar{y}^k - (1/\rho)\sum_{i=1}^{n} \nabla f_i^{1/\rho}(z_i^k)\right). \tag{A.7}$$

As presented in Section II, each agent's $\nabla f_i^{1/\rho}(z_i^k)$ is predicted by the GP and this prediction is used by the ADMM algorithm when the coordinator skips a communication round with an agent. This dynamic is expressed in (7) with the variable β_i^k, where depending on the communication decision, β_i^k takes the value of $\nabla f_i^{1/\rho}(z_i^k)$ or its predicted value. In the context of our problem, we replace $\nabla f_i^{1/\rho}(z_i^k)$ from the expressions in (A.6) and (A.7) with the dynamics defined in (7), giving the ADMM expression

$$\begin{aligned} x_i^{k+1} &= z_i^k - (1/\rho)\beta_i^k \\ u^{k+1} &= (1/\rho)\nabla h^{n/\rho}\left(n\bar{y}^k - (1/\rho)\sum_{i=1}^{n}\beta_i^k\right) \\ \bar{y}^{k+1} &= \bar{y}^k - 1/(\rho n)\sum_{i=1}^{n}\beta_i^k - u^{k+1}. \end{aligned} \tag{A.8}$$

Defining the variable $v^k = n\bar{y}^k - (1/\rho)\sum_{i=1}^n \beta_i^k$, we get that the u-update is given by

$$u^{k+1} = (1/\rho)\nabla h^{n/\rho}\left(v^k\right). \tag{A.9}$$

Appendix B: Proof of Proposition 1

Theorem 6 in [27] presents upper and lower bounds for an inequality following the format of (13). However, the derived bounds included variables that were not fully defined. For that reason, we follow the proof of Lemma 1 in [28] to get a better-defined bound for the inequality in (13).

In the proof of Lemma 1 in [28], we get that for a random vector Z with individual components $Z_l \sim \mathcal{N}(0,1)$, the logarithm of the Laplace transform of $Z_l^2 - 1$ is given by

$$\psi(u) = \log[E[\exp(u(Z_l^2 - 1))]] = -u - \frac{1}{2}\log(1-2u),$$

which for $0 < u < 1/2$ we get the bound

$$\psi(u) \leq \frac{u^2}{1-2u}.$$

Therefore, extending the previous expressions for a variable $Y = \sum_{l=1}^p a_l(Z_l^2 - 1)$, with $a_l \geq 0$, we get

$$\log[E[\exp(uY)]] = \sum_{l=1}^{p}\log\left[E[\exp(ua_l(Z_l^2-1))]\right] \leq \sum_{l=1}^{p}\frac{a_l^2u^2}{1-2a_lu}, \tag{B.1}$$

which leads to the inequality

$$\log[E[\exp(uY)]] \leq \frac{||a||_2^2u^2}{1-2||a||_\infty u}. \tag{B.2}$$

On the other hand, in [29] it was proven that if

$$\log[E[\exp(uY)]] \leq \frac{vu^2}{2(1-2cu)}, \tag{B.3}$$

then, for any positive x,

$$\mathrm{P}(Y \geq cx + \sqrt{2vx}) \leq \exp(-x). \tag{B.4}$$

Thus, given (B.2) and (B.3) we get that $v/2 = ||a||_2^2$ and $c = 2||a||_\infty$, which allow us to rewrite (B.4) as

$$\mathrm{P}(Y \geq 2||a||_\infty x + 2||a||_2\sqrt{x}) \leq \exp(-x). \tag{B.5}$$

We can define $\alpha = 2||a||_\infty$ and $\beta = 2||a||_2$, and by equalling $2||a||_\infty x + 2||a||_2\sqrt{x}$ to a positive number t we get

$$\alpha x + \beta\sqrt{x} = t$$

$$\alpha x + \beta\sqrt{x} - t = 0.$$

Solving the quadratic equation we get that

$$\sqrt{x} = \frac{-\beta + \sqrt{\beta^2 + 4\alpha t}}{2\alpha},$$

where we can obtain a value for x that depends on t and will be named $x_{(t)}$ defined as

$$x_{(t)} = \frac{\beta^2}{2\alpha^2} - \frac{\beta}{2\alpha^2}\sqrt{\beta^2 + 4\alpha t} + \frac{t}{\alpha}. \tag{B.6}$$

Introducing the definition of α and β into (B.6) we get

$$x_{(t)} = \frac{||a||_2^2}{2||a||_\infty^2} - \frac{||a||_2^2}{2||a||_\infty^2}\sqrt{1 + \frac{2t||a||_\infty}{||a||_2^2}} + \frac{t}{2||a||_\infty}, \tag{B.7}$$

which after some algebraic manipulations can be expressed as

$$x_{(t)} = \left(\sqrt{\frac{t}{2||a||_\infty} + \frac{||a||_2^2}{4||a||_\infty^2}} - \frac{||a||_2}{2||a||_\infty}\right)^2. \tag{B.8}$$

Inserting (B.8) and $\alpha x + \beta\sqrt{x} = t$ into (B.5), we get the expression for the desired probability as

$$\mathrm{P}\left[Y \geq t\right] \leq \exp(-x_{(t)}), \forall t \geq 0. \tag{B.9}$$

Going back to the context of the inequality in (13) given by

$$\mathrm{P}\left[X + \sum_{l=1}^{p}\lambda_l \geq ||\mu||_2^2\delta^2\right] \leq \xi,$$

and since $\sum_{l=1}^p \lambda_l = \mathrm{tr}(\Sigma)$ this inequality is expressed as

$$\mathrm{P}\left[X \geq ||\mu||_2^2\delta^2 - \mathrm{tr}(\Sigma)\right] \leq \xi. \tag{B.10}$$

This probability can be also bounded following (B.9) as

$$\mathrm{P}\left[X \geq ||\mu||_2^2\delta^2 - \mathrm{tr}(\Sigma)\right] \leq \exp(-x^*_{(||\mu||_2^2\delta^2 - \mathrm{tr}(\Sigma))}) \leq \xi, \tag{B.11}$$

where $x^*_{(||\mu||_2^2\delta^2-\mathrm{tr}(\Sigma))}$ is the specific form for our problem of (B.8) which is defined as

$$x^*_{(||\mu||_2^2\delta^2-\mathrm{tr}(\Sigma))} = \left(\sqrt{\frac{||\mu||_2^2\delta^2 - \mathrm{tr}(\Sigma)}{2\lambda_1} + \frac{\sum_{l=1}^p\lambda_l^2}{4\lambda_1^2}} - \frac{\sqrt{\sum_{l=1}^p\lambda_l^2}}{2\lambda_1}\right)^2, \tag{B.12}$$

with λ_l representing the eigenvalues of the covariance matrix Σ and λ_1 representing the biggest of those eigenvalues. Combining (B.11) and (B.12) we find a bound on the trace of

Σ given by

$$
\begin{aligned}
&-\left(\sqrt{\frac{||\mu||_2^2\delta^2-\text{tr}(\Sigma)}{2\lambda_1}+\frac{\sum_{l=1}^{p}\lambda_l^2}{4\lambda_1^2}}-\frac{\sqrt{\sum_{l=1}^{p}\lambda_l^2}}{2\lambda_1}\right)^2 \leq \ln(\xi)\\
&\sqrt{\frac{||\mu||_2^2\delta^2-\text{tr}(\Sigma)}{2\lambda_1}+\frac{\sum_{l=1}^{p}\lambda_l^2}{4\lambda_1^2}}-\frac{\sqrt{\sum_{l=1}^{p}\lambda_l^2}}{2\lambda_1} \geq \sqrt{\ln(1/\xi)}\\
&\frac{||\mu||_2^2\delta^2-\text{tr}(\Sigma)}{2\lambda_1}+\frac{\sum_{l=1}^{p}\lambda_l^2}{4\lambda_1^2} \geq \left(\sqrt{\ln(1/\xi)}+\frac{\sqrt{\sum_{l=1}^{p}\lambda_l^2}}{2\lambda_1}\right)^2\\
&\text{tr}(\Sigma) \leq ||\mu||_2^2\delta^2-2\left(\lambda_1\ln(1/\xi)+\sqrt{\ln(1/\xi)}\sqrt{\sum_{l=1}^{p}\lambda_l^2}\right)
\end{aligned}
\tag{B.13}
$$