Franklin Open 6 (2024) 100080

Contents lists available at ScienceDirect

Franklin Open

journal homepage: www.elsevier.com/locate/fraope

Optimal querying for communication-efficient ADMM using Gaussian process regression☆

Aldo Duarte [a,*], Truong X. Nghiem [b], Shuangqing Wei [a]

[a] Division of Electrical and Computer Engineering, School of Electrical Engineering and Commputer Science, Louisiana State University, Baton Rouge, LA 70803, United States

[b] School of Informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, AZ 86011, United States

ARTICLE INFO

Keywords:
Gaussian process
ADMM
Distributed optimization
Proximal operator
Communication reduction

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, necessitating communication reduction in scenarios where communication is costly. Integrating Gaussian processes (GP) as a learning component to the Alternating Direction Method of Multipliers (ADMM) has proven effective in learning each agent's local proximal operator to reduce the required communication exchange. A key element for integrating GP into the ADMM algorithm is the querying mechanism upon which the coordinator decides when communication with an agent is required. In this paper, we formulate a general querying decision framework as an optimization problem that balances reducing the communication cost and decreasing the prediction error. Under this framework, we propose a joint query strategy that takes into account the joint statistics of the query and ADMM variables and the total communication cost of all agents in the presence of uncertainty caused by the GP regression. In addition, 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.

1. 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 those schemes, the underlying local sub-problems in the form of *proximal minimization problems* [1] are solved by the agents in response to queries sent by the coordinator. Proximal minimization is suitable for networks with privacy constraints because it prevents each agent's local objective and constraints from being disclosed to the coordinator or other agents. Once the coordinator receives the local proximal minimization solutions from the agents, it uses them to calculate new queries for the agents that keep on driving the agents' solutions to the global solution. Such distributed optimization schemes have been applied to power management for smart buildings and distribution power systems, among other applications, as shown in [2].

The Alternating Direction Method of Multipliers (ADMM) [3] is an algorithm well suited for distributed optimization settings. It has found great success in distributed optimization due to its simplicity of implementation and its suitability for parallelization. As a result, ADMM has found many applications in machine learning problems [4] and other distributed optimization problems [5–8].

The query-response mechanism inherent to distributed optimization algorithms (ADMM included) often requires many iterations before the algorithm converges to a solution. An extensive amount of communication between the coordinator and the agents could make the system unviable in cases where communication is expensive, such as underwater communication for robot formation control [9]. For that reason, reducing communication expenditure is highly desirable, even critical, for the viability of these distributed optimization schemes in real-life applications.

Communication reduction in distributed optimization settings has previously been studied. The authors of [10] presented a hierarchical distributed optimization algorithm for the predictive control of a

☆ This material is based upon work supported by the U.S. National Science Foundation (NSF) under Grant No. 2238296.

* Corresponding author.
E-mail addresses: aduart3@lsu.edu (A. Duarte), Truong.Nghiem@nau.edu (T.X. Nghiem), swei@lsu.edu (S. Wei).

https://doi.org/10.1016/j.fraope.2024.100080
Received 28 August 2023; Received in revised form 2 February 2024; Accepted 25 February 2024
Available online 1 March 2024

smart grid with reduced communication overhead by avoiding communication between agents; however, agents communicate with the coordinator at each iteration. This is different from our proposed approach, which not only avoids communications between the agents but also skips communications between the agents and the coordinator whenever possible. Solutions for large-scale machine learning applications using distributed optimization schemes that are efficient in communication were proposed in [11,12]. These approaches are different from our work in that they do not predict the solution of the local sub-problems, but rather they turn the complex global problem into simpler sub-problems to be solved in parallel by the agents, thereby reducing communication rounds by constraining the required iterations to reach convergence. In [13], ADMM-based communication-efficient federated learning algorithms are proposed, which perform aggregation at a central coordinator of the updates sent by other agents at predefined intervals, instead of assessing the need for communication at each iteration as in this work. The authors of [14] propose using a communication censoring strategy to develop a communication-efficient ADMM algorithm to solve a convex consensus optimization problem. In contrast with our approach, the censoring mechanism reduces the communication load by recycling an agent's previous response if there is not enough variation between the agent's new response and the precedent one.

An alternative approach to communication reduction in distributed optimization via ADMM was proposed in [15] and developed with further detail in [16]. In this approach, a coordinator uses predictions from the proximal operators of local agents to skip communication with an agent whenever its corresponding prediction is deemed accurate enough. The approach uses the theory of *the Moreau envelope function* and its connections to the proximal operator [17, Chapter 1.G]. This idea was further extended in [18], where the predictions of local proximal operators and their gradients are obtained by Gaussian processes (GP). The GP models generate estimations of prediction uncertainty, which the coordinator uses to decide when communication with each agent is necessary. In our work [19], further communication reduction was achieved by extending this method to incorporate Lloyd's and uniform quantization in communications between the coordinator and agents to reduce the payload size of the shared information. We further refined our approach in [20], where a GP-based linear regression method was developed to properly account for the impact of the uniform quantization error on learning and prediction with GP.

In the above approaches, the querying mechanism to decide when a communication round should be skipped greatly affects the desired communication cost and the performance of the ADMM algorithm; therefore, developing a systematic approach for this is critical. Our work in [18] proposed a querying mechanism using a heuristic method, which decides when to communicate with an agent by comparing the conditional variance given by its corresponding GP to a threshold that adapts at each algorithmic iteration depending on the performance of the ADMM algorithm. Although this strategy was effective, it was based on an intuitive idea rather than a well-founded systematic querying mechanism. It remains unclear whether additional communication costs can be reduced using a more effective querying approach while properly solving the underlying optimization problem. This is the primary question we address in this paper.

Our main contributions in this paper are: (1) We propose a systematic querying framework to balance two criteria: reducing communication overhead and maintaining a good optimization performance. (2) We develop a joint querying method based on the general framework to make joint communication decisions for all agents. (3) We develop three simpler approximate querying strategies through which the controller makes individual decisions about when to query each agent. (4) We validate our methods through extensive simulations of a distributed sharing problem with quadratic cost functions. The simulation results show significant reductions in total communication expenditure in all test cases compared to the vanilla ADMM. Furthermore, all query methods present an acceptable trade-off between communication expenditure reduction and optimization accuracy. Lastly, the joint querying method outperforms all the other query methods in terms of their trade-off performances, as evidenced by the numerical results.

Paper Organization: We begin with the problem formulation in Section 2. The systematic querying framework is presented in Section 3. We present our proposed joint query mechanism in Section 4, followed by our proposed individual query strategies in Section 5. A probabilistic comparison between the proposed methods, which leads to an expected querying behavior, is presented in Section 6. The simulation results are presented in Section 7, and the conclusions are made in Section 8.

Notations

Let $\mathbb{R}$ denote the set of real numbers and $\mathbb{R}^p$ denote the set of p-dimensional vectors of real numbers. $\mathbb{E}[.]$ refers to the expectation operator, while $\mathrm{Cov}[.]$ is the covariance operator. The operator $\mathrm{argmin}\{f(x)\}$ returns a value of x that minimizes the function $f(x)$. $\mathrm{tr}(A)$ is the trace of the square matrix A, defined as the sum of its diagonal entries. For a p-dimensional vector $x \in \mathbb{R}^p$, $\|x\|_1$ and $\|x\|_2$ denote the L1 and L2 norms of x, respectively, while $\min_{1\le l\le p}(x_l)$ and $\max_{1\le l\le p}(x_l)$ refer to the minimum element and the maximum element of x, respectively. We consider a distributed optimization setting for multiple agents, so the subscript i of a variable refers to the i-th agent and the superscript k of a variable refers to the algorithmic iteration count.

2. Problem formulation

This paper considers a sharing problem with n agents and a central coordinator, similar to that in [4,6]. In this problem, a global cost, which includes all agents' convex local cost functions f_i: $\mathbb{R}^p \mapsto \mathbb{R}$ on local decision variables $x_i \in \mathbb{R}^p$ and a convex shared cost function h: $\mathbb{R}^p \mapsto \mathbb{R}$, is minimized, as denoted by the expression

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

The cost function f_i is known only to its corresponding agent. Additionally, the problem (1) is solved with communication allowed only between the coordinator and agents, but without exchange between agents.

The sharing problem (1) is solved using ADMM as shown in [4] with the following updates

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

where k is the algorithmic iteration count, $\rho > 0$ is a penalty parameter and $\bar{x}^k = (1/n)\sum_{i=1}^{n} x_i^k$. In iteration k, the coordinator sends a query value z_i^k to the ith agent and receives the following local *proximal operator* as a response

$$\mathbf{prox}_{\frac{1}{\rho}f_i}(z_i^k) = \underset{x_i\in\mathbb{R}^p}{\mathrm{argmin}} \left\{ f_i(x_i) + \frac{\rho}{2}\|x_i - z_i^k\|^2 \right\}. \tag{3}$$

The x-minimization step in (2) consists of the local proximal minimization problem, for every agent i,

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

2.1. STructural estimation of proximal operator with Gaussian processes (STEP-GP) overview

For brevity, we will drop the subscript i and the superscript k in the subsequent equations. The Moreau envelope of f is defined as

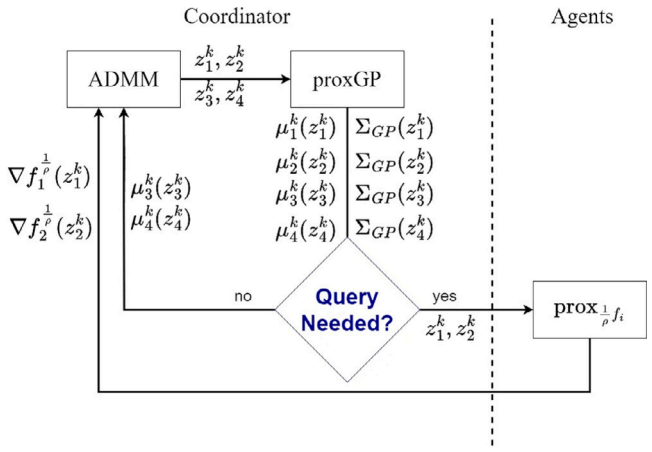

Fig. 1. Flow diagram of the query decision and the query process and response between the coordinator and 4 agents in the proposed approach.

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

When f is a convex function, the Moreau envelope $f^{\frac{1}{\rho}}$ is *convex and differentiable with Lipschitz continuous gradient with constant* ρ. Furthermore, given that 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 optimal solution of (3) only requires the gradient $\nabla f^{\frac{1}{\rho}}(z)$ to be reconstructed. In [18], we proposed using GP to learn the local proximal operators, based on the training sets from past data to predict $\nabla f^{\frac{1}{\rho}}(z)$, thus improving the STEP method in [16]. This approach is named STEP-GP.

In particular, in STEP-GP, the coordinator maintains a GP model, named proxGP, for each agent. Each GP model predicts the gradient $\nabla f_i^{\frac{1}{\rho}}(z_i^k)$ of each agent's Moreau envelope, which 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 $\mathrm{Cov}\left[\nabla f_i^{\frac{1}{\rho}}(z_i^k)\right] = \Sigma_{\mathrm{GP}}(z_i^k)$. The coordinator then uses an uncertainty measurement coming from the conditional covariance matrix to decide whether to query each agent. More details of the STEP-GP method can be found in [18].

2.2. Query-response dynamics

In Fig. 1, we present one round of the proposed algorithm for a network of 4 agents. 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 corresponding proxGP for each agent, which is trained on its past query data with the agent. The coordinator first calculates the query variables z_i^k for each agent and uses them as input to the agent's proxGP. Using the covariance matrices $\Sigma_{\mathrm{GP}}(z_i^k)$ given by the proxGPs, the coordinator decides which agents are 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 their proximal minimization problems as in (4), depicted by block $\mathbf{prox}_{\frac{1}{\rho}f_i}$. It then receives the Moreau envelopes $f_1^{\frac{1}{\rho}}(z_1^k)$, $f_2^{\frac{1}{\rho}}(z_2^k)$ and their gradients $\nabla f_1^{\frac{1}{\rho}}(z_1^k)$, $\nabla f_2^{\frac{1}{\rho}}(z_2^k)$ as responses from agents 1 and 2. Meanwhile, for agents 3 and 4, which are not queried, the coordinator uses the corresponding predicted values $\mu_3^k(z_3^k)$ and $\mu_4^k(z_4^k)$ from their proxGPs to perform the ADMM updates.

2.3. ADMM updates with GP

Following the query-response mechanism presented in Fig. 1, the ADMM expressions in (2) are modified to include the proxGP regression. First, let us 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 obtain 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 the GP. We then define the received value β_i^k as

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

The ADMM expressions in (2) can now be reformulated as:

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

This paper focuses on how the query decision-making, represented by the blue diamond block "Query Needed?" in Fig. 1, can be carried out effectively.

3. General querying decision framework

The main objective of including GP regression in the ADMM algorithm when solving a distributed optimization problem is to reduce communication overhead. However, we do not want it to significantly affect the algorithm convergence and accuracy of the optimization solution. A key component in the ADMM updates when GP is used, as presented in (8), is the variable β_i^k. This variable becomes the exact gradient $\nabla f_i^{\frac{1}{\rho}}(z_i^k)$ of the Moreau envelope or its predicted value, depending on γ_i^k. In (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. This trajectory is affected by β_i^k, which depends on the communication decision variable γ_i^k as defined in (6) and (7), which in turn affects the GP regression accuracy and the optimization performance. Therefore, the mechanism to decide γ_i^k will ultimately impact the overall communication and optimization 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 inaccurate solution upon reaching convergence. We propose a systematic querying framework that balances two opposing criteria: communication overhead reduction and optimization performance. In this framework, the querying decision solves an optimization of the form

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

where $\mathrm{comm}(\gamma^k)$ is a communication cost function, $\mathrm{uncer}(\gamma^k)$ is an uncertainty function caused by the GP regression, and ψ^k is a given threshold that fluctuates at each iteration. The uncertainty is compared with the threshold because we want to limit the prediction error at each step so that the reduction in communication does not introduce an insurmountable amount of error to the ADMM algorithm. Therefore, the decision outcomes depend on how we measure those criteria. We can define the communication cost in several ways, such as the number of agents communicating at each iteration or the number of bits exchanged at each communication round. The uncertainty is measured by the prediction uncertainty of the agents' proxGPs. Thus, we define

the query strategy in (9) as minimizing the communication cost under a constraint on the uncertainty introduced by proxGPs.

In general, the optimization problem (9) has to be solved using a combinatorial approach due to the n binary variables $\{\gamma_i^k\}_{i=1,\dots,n}$. The computation cost, therefore, could be prohibitive when the number of agents is large. For that reason, in this work, we will seek approaches for solving (9) under certain communication cost and uncertainty functions without resorting to combinatorial techniques.

4. Proposed joint query method

In this section, we propose a joint query strategy within the general framework, where the uncertainty function in (9) is the trace of the joint covariance matrix of the ADMM variables affected by the GP regression. In the following subsection, we justify why this uncertainty function is a suitable representation of the overall prediction error.

4.1. Justification of adopting trace of the covariance matrix as the uncertainty function

Consider a real Gaussian random vector $F \sim \mathcal{N}(\mu, \Sigma)$ with mean vector μ and covariance matrix Σ_F, where the l^{th} element of μ is μ_l, and the l^{th} element of F is F_l, with $l \in \{1, \dots, p\}$. Our objective is to determine a sufficient condition for the L2 norm of the discrepancy between F and its mean to be small with high probability. This can be expressed by the confidence sphere:

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

where ξ and δ are two small numbers chosen in advance for quality control. The values of δ and ξ must be small because we want the discrepancy between the actual value and the mean of F to be small with high probability, so these control variables will determine how tight we allow the discrepancy to be and with how much probability.

The following proposition presents a sufficient condition for (10).

Proposition 1. *A sufficient condition for (10) is given by*

$$\mathrm{tr}(\Sigma_F) \leq \|\mu\|_2^2 \delta^2 - 2\left(\lambda_1 \ln(1/\xi) + \sqrt{\ln(1/\xi)}\sqrt{\sum_{l=1}^{p} \lambda_l^2}\right). \tag{11}$$

Proof. The proof is presented in Appendix B.

Proposition 1 suggests that the trace of the joint covariance matrix of the ADMM variables affected by GP regression, as the random vector F, can be constrained to control the desired prediction errors, which affect the convergence of the algorithm and the accuracy of the solution. Therefore, it justifies the use of this trace as the uncertainty function $\mathrm{uncer}(\gamma^k)$ in (9).

4.2. Proposed joint query method

Following the general querying decision framework presented in Section 3, we propose using the L1 norm of γ^k as the communication cost function, which indicates how many agents are queried in the current iteration.

The uncertainty function $\mathrm{uncer}(\cdot)$ is selected based on the analysis in the previous subsection and the work [22]. The authors of [22] present a stochastic approach to inexact ADMM in which the expectation of the mean square error of the inexact ADMM variables with respect to their exact counterparts is bounded. It can be shown that the bounded expectation is equal to the trace of the error covariance matrix. Extending both analyses to our problem, we propose to use the trace of the joint covariance matrix of the iterative variables of the ADMM algorithm, given by $\mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}|\gamma^k])$, to derive the uncertainty function. Here, $\mathrm{tr}(\cdot)$ is the trace operator.

We thus have the following realization of the general optimization problem (9):

$$\begin{aligned} \text{minimize} \quad & \|\gamma^k\|_1 \\ \text{subject to} \quad & \gamma_i^k \in \{0,1\}, 1 \leq i \leq n, \\ & \mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}]|\gamma^k) \leq \psi^k, \end{aligned} \tag{12}$$

where the threshold ψ^k varies at each iteration. The rationale for (12) is to choose the smallest set of agents to query while ensuring that the trace of the joint uncertainty caused by not querying the other agents does not exceed the threshold ψ^k, thus ensuring that there is a high probability that the uncertainty is within a desired sphere. Following the convergence analysis for the stochastic inexact ADMM in [22], we choose the sequence of thresholds ψ^k such that $\sum_{k=1}^{\infty} \psi^k < \infty$. More details on ψ^k are presented in Section 4.4.

Next, we present an efficient solution to the problem in (12) without resorting to a combinatorial approach by exploiting the convexity and linearity of the cost functions and constraints considered. The idea is that the search for a set of agents to query starts with the scenario where the communication cost is maximum and the uncertainty is minimum. Then, we calculate the contribution to the joint trace of each agent where the ones 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 named *L1Norm-Trace* follows the steps listed below at iteration k:

1. For each agent, calculate its uncertainty contribution $\mathrm{un}_i = \mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}|\gamma_i^k = 0, \gamma_{j\neq i}^k = 1])$.
2. In the order from the smallest to the largest un_i, pick all the agents whose sum of un_i does not exceed the threshold ψ^k and set their γ_i^k to 0, i.e., they are not queried. The remaining agents are to be queried, i.e., their γ_i^k are set to 1.

The proposed strategy does not consider all possible combinations of communicating agents, as it would be necessary to combinatorically solve the problem posed in (12). However, our strategy solves this optimization problem optimally.

Lemma 1. *The L1Norm-Trace method solves the optimization problem in (12) optimally.*

Proof. If our method is not optimal, then our selection of agents to be queried does not minimize the communication cost while ensuring that the uncertainty constraint is met. Because we select agents from the smallest to the largest un_i, we select the largest number of agents to not be queried such that the uncertainty constraint is met. There is no other selection of agents that can further reduce $\|\gamma^k\|_1$ without violating $\sum_{i=1}^{n} \mathrm{tr}(\mathrm{Cov}[x_i^{k+1}; \bar{y}_i^{k+1}; u_i^{k+1}]|\gamma_i^k) \leq \psi^k$.

The next subsections derive the calculation of the joint trace $\mathrm{tr}(\mathrm{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}]|\gamma^k)$ and present the mechanism to vary the threshold ψ^k.

4.3. Derivation of the trace of the ADMM joint covariance matrix

In this subsection, we first present an equivalent expression to the ADMM updates presented in (8) that allows us to see the inherent coupling of the agents. 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 notation presented in the problem definition in Section 2.

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

$$x_i^{k+1} = z_i^k - (1/\rho)\beta_i^k$$
$$u^{k+1} = (1/\rho)\nabla h^{n/\rho}\left(v^k\right)$$
$$\bar{y}^{k+1} = \bar{y}^k - 1/(\rho n)\sum_{i=1}^{n}\beta_i^k - u^{k+1}, \tag{13}$$

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 (13) presents the ADMM updates in terms of the gradient of the Moreau Envelope of functions $\{f_i\}$ and h, and follows the calculations for the ADMM algorithm executed on 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)) comes from the exact value or the predicted value of $\nabla f_i^{\frac{1}{\rho}}(z_i^k)$, so the ADMM updates in (13) can be used to quantify the joint uncertainty of the ADMM variables.

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 (13), the definition of β_i^k in (7), and that only the terms including β_i^k contribute to the uncertainty, the overall trace function becomes

$$\mathrm{tr}(\mathrm{Cov}[x^{k+1};\bar{y}^{k+1};u^{k+1}|\gamma^k]) = (1+1/n^2)(1/\rho)^2\sum_{i=1}^{n}(1-\gamma_i^k)\mathrm{tr}\left(\Sigma_{\mathrm{GP}}(z_i^k)\right) + 2(1/\rho)^2\mathrm{tr}\left(\mathrm{Cov}[\nabla h^{n/\rho}(v^k)]\right), \tag{14}$$

which is subject to the function h. Calculating the covariance matrix of $\nabla h^{n/\rho}(v^k)$ given the probability distribution of v^k is generally difficult and may not have a closed-form equation, because $\nabla h^{n/\rho}(\cdot)$ is generally a nonlinear function. In this case, we must approximate this covariance matrix [23]. However, this approximation will introduce uncertainty, which will propagate into the algorithmic iterations, affecting the communication decision methods and having an impact on the ADMM algorithm.

4.4. Threshold ψ^k mechanism

During the execution of the ADMM algorithm, the uncertainty of the GP regression tends to reduce when the ADMM algorithm gets closer to convergence. This is because more training data from responses to queries is available, which allows 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.

$$\psi^{k_0} = \iota V^{k_0}, \tag{15}$$

where V^{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, chosen in advance, is a number between 0 and 1. Given a preselected decay rate $\alpha \in (0,1)$, at a later iteration $k > k_0$, the threshold is updated as:

$$\psi^k = \psi^{k_0}\alpha^{k-k_0}. \tag{16}$$

5. Proposed individual query methods

In this section, we simplify the query framework presented in Section 3 by proposing three individual query methods to determine when a communication round between the coordinator and the 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 uncertainty measures of the other agents. This strategy reduces considerably the computational complexity of the general method presented in Section 3, 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 ensure that the prediction error does not affect the ADMM's algorithm performance greatly. Although this approach is not as rigorous as the joint method, its simplicity makes it suitable for applications where the computational cost must be as low as possible.

In an individual query method, the decision is made per agent where this 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 coordinator shall decide 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 ways. By doing this, we drop the minimization problem presented in (9) and set each γ_i^k by comparing the estimated error of each agent to a threshold individually. The individual query strategies proposed were not arbitrarily derived, but followed 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 following subsections.

5.1. Maximum variance query method

Similarly to the derivation presented in Section 4.1, our goal is to generate a decision rule in which the prediction error is small with a high probability. For that reason, using the concept of confidence interval, a threshold setting can be derived. When the prediction error is below a chosen threshold, no query will be sent to an agent. As a consequence, we want the probability that the estimation error is bounded by a small upper bound to be as large as possible.

For the following derivations, we employ the general notation used in Section 4.1, where the 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 $s^2 = \mathrm{diag}(\Sigma_F)$, where the l^{th} element of s^2 is s_l^2. The desired confidence interval is given by

$$\mathrm{P}\left[-\delta\|\mu\|_1 \le \|F-\mu\|_1 = \sum_{l=1}^{p}|F_l-\mu_l| \le \delta\|\mu\|_1\right] \ge 1-\xi, \tag{17}$$

A sufficient condition of (17) is given below in terms of the requirement imposed on each dimension F_l of F.

$$\mathrm{P}\left[\left|\frac{F_l-\mu_l}{s_l}\right| \le \frac{\delta|\mu_l|}{s_l}, 1 \le l \le p\right] \ge 1-\xi. \tag{18}$$

Following the region probability defined in [24], we get an immediate bound of (18):

$$\mathrm{P}\left[\left|\frac{F_l-\mu_l}{s_l}\right| \le \frac{\delta|\mu_l|}{s_l}, 1 \le l \le p\right] \ge \prod_{l=1}^{p}\mathrm{P}\left[\left|\frac{F_l-\mu_l}{s_l}\right| \le \frac{\delta|\mu_l|}{s_l}\right]. \tag{19}$$

and it implies that if the following condition holds true,

$$\mathrm{P}\left[\left|\frac{F_l-\mu_l}{s_l}\right| \le \frac{\delta|\mu_l|}{s_l}\right] \ge 1-\xi', \forall 1 \le l \le p, \tag{20}$$

where $1-\xi' = (1-\xi)^{1/p}$, the requirement in (17) is immediately satisfied.

However, instead of analyzing this condition for each of the dimensions of F, we can simplify the analysis by further requiring that the maximum standard deviation (the maximum element of the vector s) satisfy the condition inside the probability in (18) when the bound is minimum. This is achieved when

$$\mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \le \frac{\delta\min_{1\le l\le p}|\mu_l|}{\max_{1\le l\le p}(s_l)}\right] \ge 1-\xi', \ \forall 1 \le l \le p, \tag{21}$$

The condition in (20) is met when requiring

$$\max_{1\le l\le p}(s_l) \le \frac{\min_{1\le l\le p}|\mu_l|\delta}{Q^{-1}(\xi'/2)} = \psi^{(1)}, \tag{22}$$

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 (22) can be used as the threshold $\psi^{(1)}$ to compare the maximum element of the vector of variances s (s_l). In case $\max_{1\le l\le p}(s_l) \le \psi^{(1)}$, then automatically all the elements of s satisfy the condition.

In the context of the problem defined in Section 2, at each iteration, the GP regression gives us for agent i the predicted mean $\mu_i^k(z_i^k)$ and the conditional covariance matrix $\Sigma_{\text{GP}}(z_i^k)$. In this scenario, the vector of variances will be defined as $(s_i^k)^2 = \text{diag}(\Sigma_{\text{GP}}(z_i^k))$. Furthermore, as mentioned in the previous section, each agent's GP prediction uncertainty is reduced over the algorithmic rounds. For that reason, the threshold $\psi^{(1)}$ should not be static as also implied in (22) but should decrease over the ADMM iterations. This requires the control variables ξ and δ to be adjusted 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 (22), but instead employ a general threshold ψ_i^k per agent that follows the threshold mechanism described in Section 4.4. Finally, under this querying mechanism, the variable γ_i^k is defined as

$$\gamma_i^k = \begin{cases} 0, & \text{if } \max_{1\le l\le p}(s_{i[l]}^k) \le \psi_i^k \\ 1, & \text{otherwise.} \end{cases} \tag{23}$$

5.2. Maximum variance and mean ratio query method

The subsequent proposed strategy expands from the confidence interval analysis presented in Section 5.1 to build its mathematical intuition. Following the confidence interval defined in (18), to require that each dimension of an agent has a small relative estimation error, we are interested in evaluating the bound in (19). Defining $a^* = \max_{1\le l\le p}\frac{s_l}{|\mu_l|}$, it is then straightforward to show that if

$$\prod_{l=1}^{p} \text{P}\left[\left|\frac{F_l-\mu_l}{s_l}\right| \le \frac{\delta|\mu_l|}{s_l}\right] \ge \left(\text{P}\left[\left|\frac{F_l-\mu_l}{s_l}\right| \le \frac{\delta}{a^*}\right]\right)^p \ge 1-\xi, \tag{24}$$

we always satisfy

$$\text{P}\left[\left|\frac{F_l-\mu_l}{s_l}\right| \le \frac{\delta|\mu_l|}{s_l}, 1\le l\le p\right] \ge 1-\xi. \tag{25}$$

Note that under the GP model, each F_l is Gaussian following $\mathcal{N}(\mu_l, s_l^2)$, suggesting $\frac{F_l-\mu_l}{s_l}$ following $\mathcal{N}(0,1)$. We then obtain a sufficient condition to meet the confidence region requirement stated in (25), namely,

$$\max_{1\le l\le p}\frac{s_{[l]}}{|\mu_{[l]}|} \le \frac{\delta}{Q^{-1}(1/2 - 1/2 * (1-\xi)^{1/p})} = \psi^{(2)}. \tag{26}$$

The upper-bound expressed in (26) is not imposed on the maximum element of s but on the maximum ratio of $\frac{s_l}{|\mu_l|}$.

In the context of our problem defined in Section 2, the threshold $\psi^{(2)}$ should decrease over the ADMM algorithmic rounds to keep up with the reduction of the uncertainty of the GP prediction. Similarly to the query method presented in Section 5.1, we do not use the specific threshold $\psi^{(2)}$ defined in (26), but instead employ a general threshold ψ_i^k per agent following the mechanism described in Section 4.4. Using the notation of our problem, the variable γ_i^k under this query strategy is defined as

$$\gamma_i^k = \begin{cases} 0, & \text{if } \max_{1\le l\le p}\frac{s_{i[l]}^k}{|\mu_{i[l]}^k(z_i^k)|} \le \psi_i^k \\ 1, & \text{otherwise.} \end{cases} \tag{27}$$

5.3. Ratio of maximum eigenvalue and the norm of the mean query method

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 4.4. Our objective is to fulfill the decision criterion presented in (10) given by:

$$\text{P}\left[\|F-\mu\|_2 \le \|\mu\|_2\delta\right] \ge 1-\xi.$$

Following the same transformation presented in Appendix B expressed in (B.1), we seek an alternative sufficient condition to satisfy the confidence sphere condition in (B.1). We find an alternative lower bound on this probability by defining $\lambda_1 = \max_{1\le l\le p}\lambda_l$ (the maximum eigenvalue of the matrix Σ_F) and resorting to the following inequality

$$\sum_{l=1}^{p}\frac{G_l^2}{\lambda_l} \ge \frac{1}{\lambda_1}\sum_{i=1}^{p}|G_l|^2 = \frac{1}{\lambda_1}\|G\|^2, \tag{28}$$

where $G_l/\sqrt{\lambda_l}$ are independent and identical distributed (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 (B.1), we have a sufficient condition to satisfy (10) given by:

$$\text{P}\left[\|G\|_2 \le \|\mu\|_2\delta\right] \ge \text{P}\left[\sum_{l=1}^{p}\frac{G_l^2}{\lambda_l} \le \frac{1}{\lambda_1}\|\mu\|_2^2\delta^2\right] \ge 1-\xi. \tag{29}$$

This expression can be satisfied if λ_1 satisfies the following condition:

$$\frac{\lambda_1}{\|\mu\|_2^2} \le \frac{\delta^2}{\mathcal{F}_{\chi^2}^{-1}(1-\xi)} = \psi^{(3)}, \tag{30}$$

where $\mathcal{F}_{\chi^2}^{-1}(.)$ is the inverse function of the Cumulative Distribution Function (CDF) of the chi-square random variable. Thus, if $\frac{\lambda_1}{\|\mu\|_2^2} \le \psi^{(3)}$, we ensure that the confidence sphere criterion in (B.1) is met; therefore, 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 (26), we need to compare the relationship between the maximum eigenvalue and the square of the L2 norm of the conditional mean to a threshold subject to the chi-square distribution, under the confidence sphere setting. Once again, the specific threshold presented in this subsection is replaced by a general threshold ψ_i^k per agent following the mechanism described in Section 4.4. Finally, we define a query strategy in which the variable γ_i^k is defined as

$$\gamma_i^k = \begin{cases} 0, & \text{if } \frac{\lambda_1^k}{\|\mu_i^k(z_i^k)\|_2^2} \le \psi_i^k \\ 1, & \text{otherwise.} \end{cases} \tag{31}$$

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.

6. Probability comparison between querying strategies

In this section, we present a comparative analysis of the probabilities used as a basis for the various querying strategies proposed. 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 methods' probabilities first defined in Section 4.1.

6.1. Relationship between maximum variance and maximum ratio methods

Comparing the conditions presented in (20) and (24), while acknowledging the bound presented in (19), we can observe that the condition in (20) is more likely to occur. Thus, we find that the relationship between the Maximum Variance and Maximum Ratio between variance and mean methods is given by

$$\left(\mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \le \frac{\delta}{a^*}\right]\right)^p \le \mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \le \frac{\delta \min_{1\le l\le p}|\mu_l|}{\max_{1\le l\le p}(s_l)}, 1\le l\le p\right]. \tag{32}$$

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.

6.2. Relationship between L2 norm-based methods and a L1 norm condition

The querying strategies involving the maximum eigenvalue and the trace, presented in Sections 5.3 and 4.1, respectively, are derived from the same confidence sphere involving the L2 norm of $F-\mu$. This confidence region is defined in Eq. (10). We want to find a relationship between this confidence sphere and a condition involving the L1 norm of $F-\mu$ given by

$$\mathrm{P}\left[\|F-\mu\|_1 \le \delta\|\mu\|_2\right] \ge 1-\xi. \tag{33}$$

We know that for any real vector x, the relationship between L1 and L2 norms is given by $\|x\|_1 \ge \|x\|_2$. This implies

$$\mathrm{P}\left[\|F-\mu\|_1 \le \delta\|\mu\|_2\right] < \mathrm{P}\left[\|F-\mu\|_2 \le \delta\|\mu\|_2\right], \tag{34}$$

which suggests that if the condition in (33) holds true then automatically the condition in (10) is also true, thereby the querying condition based on L1 norm is more demanding than that under the L2 norm, thereby resulting more frequent queries accordingly.

6.3. Relationship between maximum variance method and an L1 norm condition

The probability in the condition given in (33) can be expressed as

$$\mathrm{P}\left[\sum_{l=1}^{p}|F_l-\mu_l| \le \delta\|\mu\|_2\right] \ge 1-\xi. \tag{35}$$

Since a sufficient condition of $\sum_{l=1}^{p}|F_l-\mu_l| \le \delta$ is $|F_l-\mu_l| \le \frac{1}{p}\delta\|\mu\|_2$, for $1\le l\le p$, we have

$$\mathrm{P}\left[|F_l-\mu_l| \le \frac{1}{p}\delta\|\mu\|_2,\ 1\le l\le p\right] \le \mathrm{P}\left[\|F-\mu\|_1 \le \delta\|\mu\|_2\right]. \tag{36}$$

Now, we want to compare the left-hand side of (36) with the probability expression for the Maximum Variance method in (18). Since the variable δ, used throughout all derived probabilities, is a variable that can be tuned, we can define a variable $\hat{\delta}$ such that $\frac{1}{p}\hat{\delta}\|\mu\|_2 = \delta\min_{1\le l\le p}|\mu_l|$. Dividing by s_l into both sides of the arguments in the probability of the left side of (36), it is straightforward to see that the following inequalities hold.

$$\mathrm{P}\left[\frac{|F_{[l]}-\mu_{[l]}|}{s_l} \le \frac{\delta\min_{1\le l\le p}|\mu_l|}{\max_{1\le l\le p}(s_l)}, 1\le l\le p\right] \le \mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \le \frac{1}{p}\frac{\hat{\delta}\|\mu\|_2}{s_l},\ 1\le l\le p\right] \le \mathrm{P}\left[\|F-\mu\|_1 \le \hat{\delta}\|\mu\|_2\right]. \tag{37}$$

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.

6.4. Complete relationship between all methods

Combining the inequalities obtained in (32), (34), and (37) with the definition of $\hat{\delta}$, we get the following inequalities

$$\begin{aligned}&\left(\mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \le \frac{\delta}{a^*}\right]\right)^p \\ &\le \mathrm{P}\left[\frac{|F_l-\mu_l|}{s_l} \le \frac{\delta\min_{1\le l\le p}|\mu_l|}{\max_{1\le l\le p}(s_l)}, 1\le l\le p\right] \\ &\le \mathrm{P}\left[\|F-\mu\|_1 \le \hat{\delta}\|\mu\|_2\right] \\ &\le \mathrm{P}\left[\|F-\mu\|_2 \le \hat{\delta}\|\mu\|_2\right].\end{aligned} \tag{38}$$

The relationships in (38) demonstrate how the probabilities used in our proposed decision strategies are related to each other. They reveal that the query 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 directly based on the L1 and L2 norm-based confidence spheres will end up with a more relaxed querying dynamics.

The following section presents numerical results to validate and compare all the proposed query methods. We will present comparisons made in terms of querying dynamics, which will be shown consistent with the analysis presented in this section and their resulting convergence speed and qualities in solving a distributed ADMM optimization problem.

7. Numerical simulations

In this section, we evaluate the proposed query methods through a numerical study of solving a sharing problem where each agent's local function is quadratic.

The details of our problem setting are presented next.

7.1. Quadratic sharing problem

7.1.1. Problem definition

We evaluate our methods using a sharing problem motivated by the application in [6]. However, we do not consider the dynamic behavior of the variables as in [6] but assume that they are stationary. The sharing problem is formulated as

$$\begin{aligned}\text{minimize}\quad &\sum_{i=1}^{n}[(1/2)x_i^T M_i x_i + w_i^T x_i + c_i] \\ &+ (1/2)\sum_{i=1}^{n} y_i^T M_h \sum_{j=1}^{n} y_j + w_h^T \sum_{i=1}^{n} y_i + c_h \\ \text{subject to}\quad &x_i - y_i = 0,\end{aligned} \tag{39}$$

where for $i=1,\dots,n$, 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.

7.1.2. Problem parameters generation

The problem's parameters presented (39) are generated following the example given in [6]. First, the parameters 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 parameter $w_i^{[0]}$ 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 parameter w_h is generated following the same procedure as w_i, but is calculated only once and not for each agent.

On the other hand, to generate M_i for each agent, we first generate a symmetric $p\times p$ matrix $M_i^{[0]} = AA'$, 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 $S_i = BB'$ is a symmetric $p \times p$ matrix with the entries of B randomly generated and uniformly distributed on [−1,1]. Subsequently, M_i is constructed as

$$M_i = \begin{cases} \tilde{M}_i, & \text{if } \lambda_{min}(\tilde{M}_i) > \epsilon_d \\ \tilde{M}_i + \left(\epsilon_d - \lambda_{min}(\tilde{M}_i)\right) I_p, & \text{otherwise,} \end{cases} \tag{40}$$

where $\lambda_{min}(\tilde{M}_i)$ denotes the smallest eigenvalue of $\tilde{M}_i$ and $\epsilon_d > 0$ is a positive constant. The parameter M_h is generated following the same procedure as M_i, but it is calculated only once and not for each agent.

7.1.3. Solution using ADMM

Following the specifics of the problem in (39) and the expression of ADMM in (2), we can derive a closed-form solution for updating the ADMM 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 equating it to zero, we obtain

$$x_i^{k+1} = (M_i + \rho I_p)^{-1}(\rho z_i^k - w_i), \tag{41}$$

where I_p is the $p\times p$ identity matrix. The expression in (41) is the closed-form solution of the optimization for the x_i update to be computed on 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

$$\bar{y}^{k+1} = (nM_h + \rho I_p)^{-1}(\rho(u^k + \bar{x}^{k+1}) - w_h). \tag{42}$$

Finally, combining the ADMM expression in (2) with the expressions in (41) and (42), the ADMM updates are expressed as

$$\begin{aligned} x_i^{k+1} &= (M_i + \rho I_p)^{-1}(\rho z_i^k - w_i) \\ \bar{y}^{k+1} &= (nM_h + \rho I_p)^{-1}((\rho/n)v^k - w_h) \\ u^{k+1} &= (1/n)(v^k - n\bar{y}^{k+1}), \end{aligned} \tag{43}$$

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

7.2. Equation of the trace of the joint covariance matrix

As presented in Section 4.2, our proposed joint 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 (43), is given by

$$\begin{aligned} &\text{tr}(\text{Cov}[x^{k+1}; \bar{y}^{k+1}; u^{k+1}|\gamma^k]) = \\ &(1 + 1/n^2)(1/\rho)^2 \sum_{i=1}^n (1 - \gamma_i^k)\text{tr}\left(\Sigma_{\text{GP}}(z_i^k)\right) + \\ &(2/n^2)\sum_{i=1}^n (1 - \gamma_i^k)\text{tr}\left(C^T C \Sigma_{\text{GP}}(z_i^k)\right) - \\ &2(1/n^2\rho)\sum_{i=1}^n (1 - \gamma_i^k)\text{tr}\left(C\Sigma_{\text{GP}}(z_i^k)\right), \end{aligned} \tag{44}$$

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

7.3. Simulation implementation

The problem in (39) is solved with two different algorithms:

1. *Sync*: this algorithm uses ADMM with proximal operator as in (2), which simplifies to (43) 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 5.1.
- *MaxRat*: The query strategy presented in Section 5.2.
- *MaxEig*: The query strategy presented in Section 5.3.
- *L1Norm-Trace*: The query strategy presented in Section 4.2.

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 two cases where the number of agents is taken from $n \in \{10, 30\}$.

Our results were generated using MATLAB. For comparison purposes, ground truth solutions to minimization problems (39) were obtained using the YALMIP toolbox [25]. For the construction of the GP models, we used the GPstuff toolbox [26]. All calculations were performed on high-performance computers at Louisiana State University (http://www.hpc.lsu.edu).

7.4. Metrics and considerations

7.4.1. Media access control (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 [27], 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 is the total number of transmissions performed. After running the simulation offline 100 times, an average efficiency ζ is obtained. At iteration k, the coordinator receives a certain number of simultaneous responses that are expressed in the variable T_{simul}^k. The expected transmission time in one iteration round will be $T_{\text{round}}^k = T_{\text{simul}}^k/\zeta^*$, where ζ^* is the average efficiency in the MAC simulation for the given scenario. The total transmission time is $\text{Tx}_t = \sum_{k=1}^N T_{\text{round}}^k$, where N is the iteration number where convergence was reached.

7.4.2. 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 that compare the primal and dual of ADMM against two different tolerances. Expressing the primal and dual in terms of the specifics of our problem results in a termination criterion of the form:

$$\|\bar{x}^{k+1} - \bar{y}^{k+1}\|_2 \le \epsilon^{\text{pri}} \text{ and } \|\rho(\bar{y}^{k+1} - \bar{y}^k)\|_2 \le \epsilon^{\text{dual}}, \tag{45}$$

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 criteria, such as

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

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 L2 norms are in $\mathbb{R}^p$. Both ϵ^{abs} and ϵ^{rel} are set manually 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.

7.4.3. 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. Define the negative logarithm of the relative error ($NLRE$) expression as

$$NLRE = -\log(|J_{gt} - J_*|/J_{gt}), \tag{46}$$

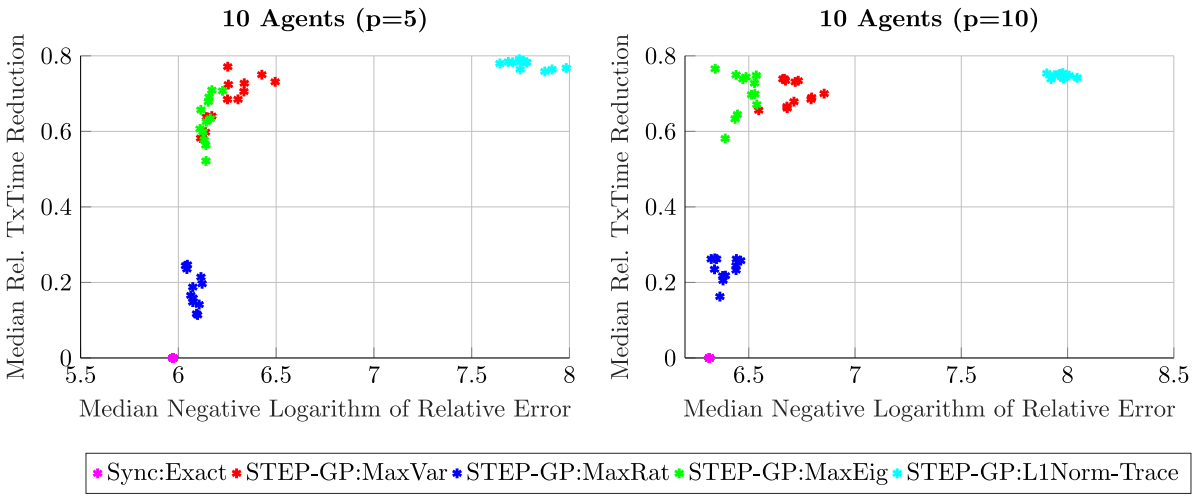

Fig. 2. Performance trade-off between the Relative Transmission Time Reduction and the Negative Logarithm of the Relative Error for 10 Agents with variable's dimension $p = 5$ (left) and $p = 10$ (right). The plots show the 12 best-ranked tuple medians of the 100 simulations for different sets of parameters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

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 us define the relative transmission time reduction (RTx) as

$$RTx = (\text{Tx}_{ADMM} - \text{Tx}_{GP})/\text{Tx}_{ADMM}, \tag{47}$$

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 the *Sync:Exact* and *STEP-GP* methods use the same set of problem parameters.

We present our results in a graph where the vertical axis shows the values of RTx and the horizontal axis shows the values of $NLRE$. Each point in the graph is a tuple of transmission time reduction and accuracy, and its location shows 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 as large as possible. Hence, we want the points to be as close to the right upper corner of the graph as possible.

7.5. Initial threshold tuning

Since the variation of the initial threshold affects the overall performance of the tested algorithms, we propose fine-tuning 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 4.4 initializes its initial threshold ψ^{k_0} following the expression in (15). Such an initialization requires manually setting the variable ι, which indicates how proportional regarding V^{k_0} we want ψ^{k_0} to be. For all the different methods tested in this chapter, we tune ψ^{k_0} considering $\iota = [0.5, 0.6 \dots, 1.4, 1.5]$.

7.6. Simulation results setting

In this subsection, we present the results for 10 and 30 agents when using the different query strategies proposed in this work with the threshold mechanism described in Section 4.4. We consider different initial threshold values following the description in Section 7.5. 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 7.1.2. 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 7.4.3.

The decaying threshold described in Section 4.4 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 tuning the initial threshold. Since we consider a set of 11 initial thresholds per method, each scenario tested 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 Figs. 2–3 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 values obtained 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 $\alpha = [0.95, 0.96, \dots, 0.99]$, and $x_i^0 = \bar{y}^0 = u^0 = 0$.

7.7. Simulation results for 10 and 30 agents

Figs. 2–3 (left) present the graph $NLRE$ vs. RTx for 10 and 30 agents of the median of 100 simulations for the *Sync:Exact* and the *STEP-GP* based algorithms for the different initial thresholds considered, per each of the values considered of α when the dimension of the variables is $p = 5$, while Figs. 2–3 (right) show the same information but when the dimension of variables is $p = 10$. The presented results were selected as a consequence of a ranking of the best points in terms of the trade-off between all values tested of α. The results in all cases show three main clusters of the points presented. In the lower-left corner, the points that show the worst performance in terms of the trade-off between communication reduction and accuracy appear, which corresponds to the *STEP-GP:MaxRat* method. In the upper-left corner, with results similar to each other in all cases, appear *STEP-GP:MaxVar* and *STEP-GP:MaxEig*. These methods present a similar reduction in 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, separated from the other methods appears *STEP-GP:L1Norm-Trace* with all its points close to each other in all the graphs presented.

On the other hand, the results presented in terms of the reduction in relative transmission time in Figs. 2–3 correlate with the analysis presented in Section 6. As the graphs show, *STEP-GP:MaxRat* presents the lowest communication reduction in all cases. The observation of the

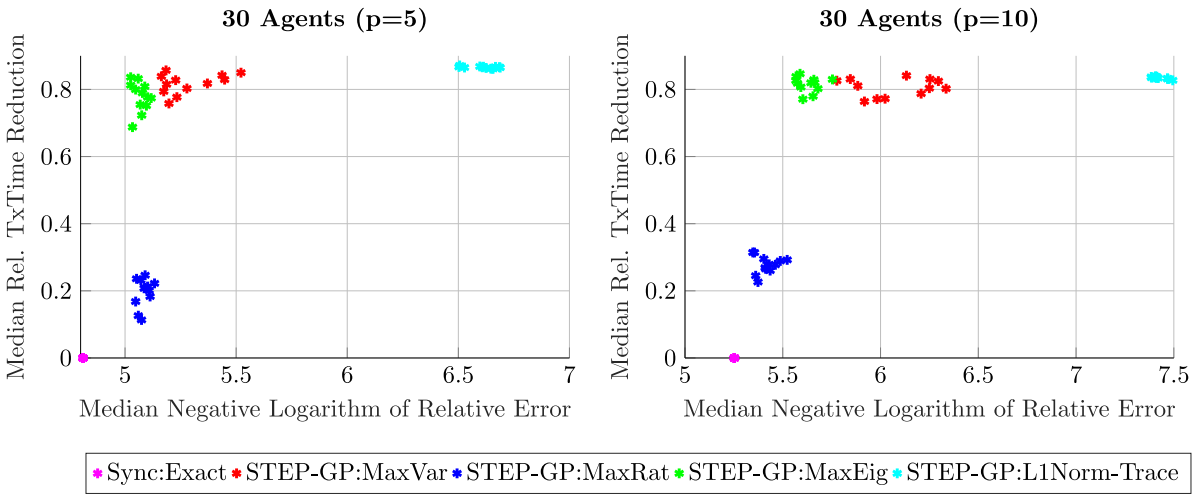

Fig. 3. Performance trade-off between the Relative Transmission Time Reduction and the Negative Logarithm of the Relative Error for 30 Agents with variable's dimension $p = 5$ (left) and $p = 10$ (right). The plots show the 12 best-ranked tuple medians of the 100 simulations for different sets of parameters M_i, M_h, w_i, w_h, c_i and c_h, and for different values of α.

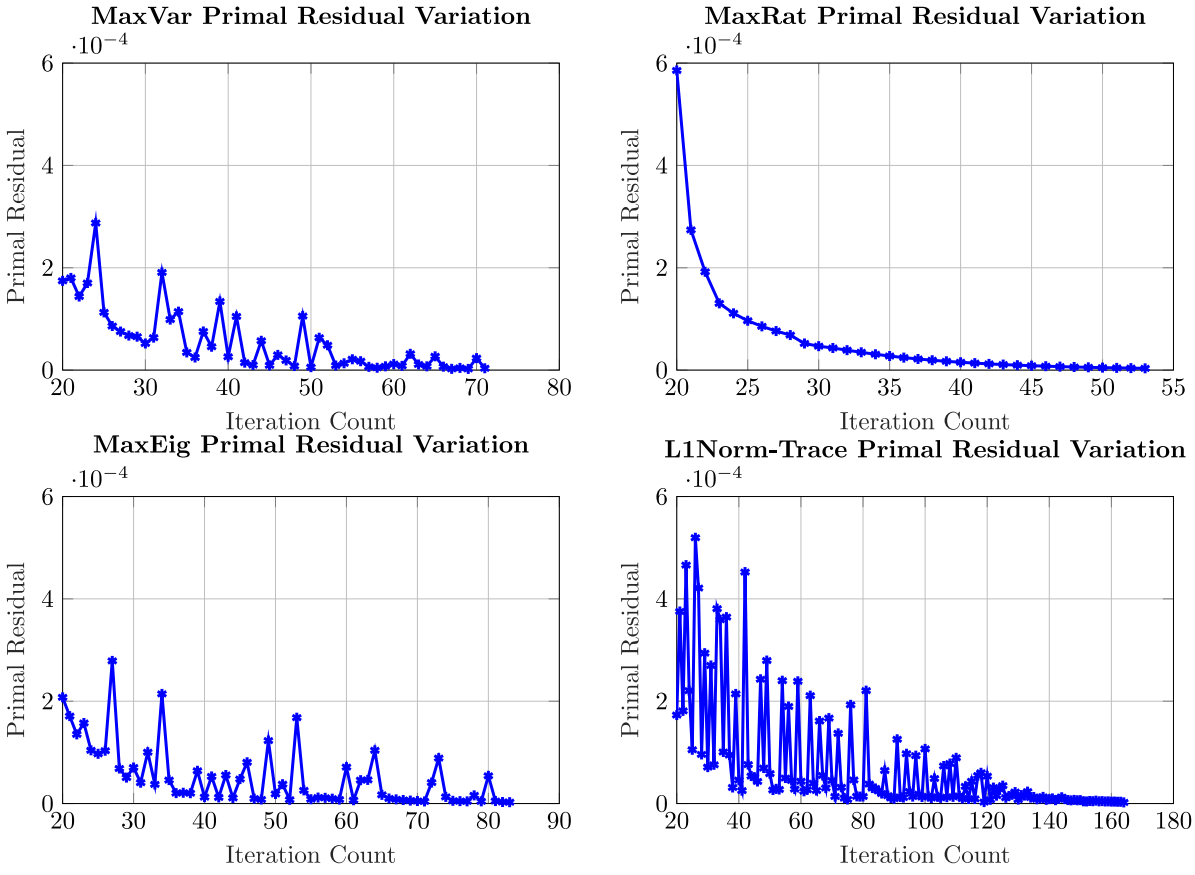

Fig. 4. Variation of the primal residual through the iteration count for all the proposed query methods. The graphs present the test scenario for the same set of parameters M_i, M_h, w_i, w_h, c_i, and c_h of 10 agents with variables' dimension of $p = 10$, an initial threshold given by $\iota = 1$, and decay rate $\alpha = 0.97$ for all cases.

intermediate results showed that this method asked queries for each agent in around 80% of the total iterations required to reach convergence. Furthermore, the two methods based on an L2 norm confidence sphere (*STEP-GP:MaxEig* and *STEP-GP:L1Norm-Trace*) present a little more reduction in relative transmission time than the *STEP-GP:MaxVar* method. This difference is not significant if we only analyze the relative transmission time reduction metric. However, through 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 the number of iterations is also very significant. Thus, the results generated are aligned with the anticipated query behavior.

7.8. Empirical convergence

In this subsection, we present results on the convergence behaviors of the proposed query methods. Fig. 4 shows the ADMM primal residual

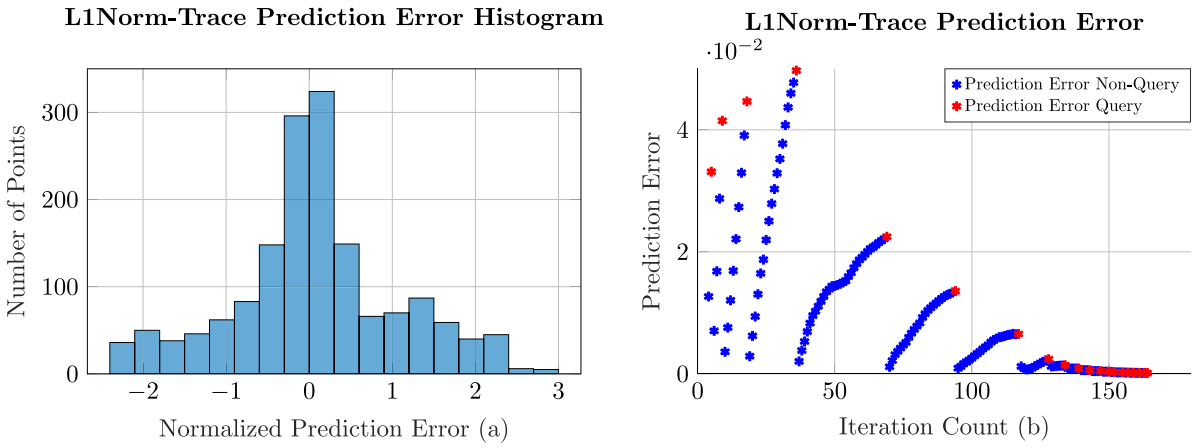

Fig. 5. Prediction Error statistics corresponding to agent 1 under the *STEP-GP:L1Norm-Trace* query strategy for a specific set of parameters M_i, M_h, w_i, w_h, c_i, and c_h in a system of 10 agents with variables' dimension of $p = 10$, an initial threshold set by $\iota = 1$, and decay rate $\alpha = 0.97$. 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.

as defined in Section 7.4.2 through the iteration count until convergence is reached for all methods tested. 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 the dimension of the variables $p = 10$, an initial threshold set by $\iota = 1$ and the decay rate $\alpha = 0.97$ for all cases. The figures presented 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 a frequency, the larger the convergence speed. The speed of convergence shown in Fig. 4 for each method is aligned with the analysis presented in Sections 6 and 7.7 because we see that *STEP-GP:L1Norm-Trace* requires considerably more iterations to reach convergence than the rest of the methods, while *STEP-GP: MaxRat* requires fewer iterations than all other methods. Although only one case is presented, this trend is observed in all test scenarios considered in all our experiments presented in the previous subsections. Thus, all generated simulations (regardless of the parameters of the test scenario) reached convergence and each query strategy presents the same convergence speed behavior.

7.9. Prediction error

In this subsection, we present statistics about how the prediction error behaves in our algorithm through all different query methods. Fig. 5 presents two graphs showing information on the prediction error of a simulation 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 the dimension of the variables $p = 10$, 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 Fig. 5 (a) we present the histogram of the normalized prediction error vector ($\epsilon^k_{i(NPE)}$), where the l^{th} entry ($l \in [1, \dots, p+1]$) is defined as

$$\epsilon^k_{i[l](NPE)} = \left(\frac{1}{s^k_{i[l]}}\right)\left|\left[f_i^{\frac{1}{\rho}}(z^k_i); \nabla f_i^{\frac{1}{\rho}}(z^k_i)\right]_{[l]} - \mu^k_{i[l]}\right|.$$

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 $\epsilon^k_{i(NPE)}$ as a point to be considered in the graph. Following the GP assumptions, we should expect that the discrepancy between the Moreau Envelope and its gradient with the predicted mean follows a Gaussian distribution. However, the histogram in Fig. 5 (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 discrepancies with respect to the expected Gaussian bell shape than the one presented in Fig. 5 (a). This is interesting because these results show that even though the assumed Gaussian distribution of $f_i^{\frac{1}{\rho}}(z^k_i)$ does not hold, the GP is still capable of making a good prediction with acceptable accuracy. Furthermore, this discrepancy from the initial assumption did not prevent any of the scenarios tested from reaching convergence.

On the other hand, Fig. 5 (b) presents the variation of the L2 norm of the prediction error at each iteration for agent 1. This is defined as

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

This metric generates a single point per iteration, so the presented graph shows the variation of the prediction error over the algorithmic iterations. Fig. 5 (b) also makes a differentiation between iterations in which a query was made (green points) and iterations in which no query was made (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 prediction error behavior is observed for all agents through all the different test scenarios and different query strategies.

7.10. Query dynamics

In this subsection, we present information on the distances between the queries z^k_i generated at each iteration compared to the previous query points included in the GP training set. Fig. 6 (a) presents the measurement of the minimum distance between a new query vector and all query vectors already in the training set. This distance is defined as

$$d(z^k_i, Z^k) = \min\{d(z^k_i, z) : z \in Z^k_i\},$$

where Z^k_i is the set containing the queries within the GP training set for agent i until iteration k and $d(\cdot)$ is the distance function. Since each generated z^k_i is a vector, the distance function considered is $d(z^k_i, Z^k) = \|z^k_i - z\|_2$ where $z \in Z^k_i$. Fig. 6 (a) presents a differentiation between iterations in which a query was made (green points) and iterations

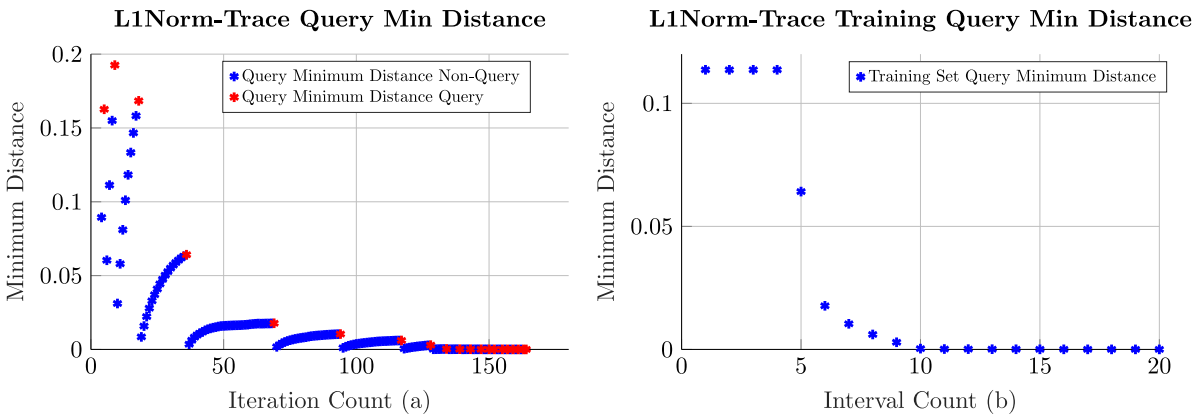

Fig. 6. Distances between generated query points for a specific set of parameters M_i, M_h, w_i, w_h, c_i, and c_h in a system of 10 agents with variables' dimension of $p = 10$, an initial threshold set by $\iota = 1$, and decay rate $\alpha = 0.97$. Graph (a) presents the measurement of the minimum distance between a new query vector and all 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.

in which there was no query (blue points). The results show that the distance between the queries throughout the iterations tends to become smaller as the iteration process approaches convergence. This is correlated with the patterns observed in Fig. 5 (b), where the prediction error is smaller when the algorithm is closer to convergence. The closer the query points are to the end of the algorithm run, the more points are trained in GP around a close vicinity, thus considerably reducing the uncertainty of the prediction. Furthermore, the behavior of the minimum distance between queries presented in Fig. 6 (a) presents a similar bursting behavior to that observed for the prediction error in Fig. 5 (b).

On the other hand, Fig. 6 (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 is this minimum distance 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 Fig. 6 (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. Similarly to the results presented in Fig. 6 (b), the distance between the query points also decreases 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.

7.11. 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. Compared to the other methods tested, for similar values of total transmission time, the *STEP-GP: L1Norm-Trace* method usually produces a global ADMM solution closer to the true solution. In contrast, the *STEP-GP:MaxRat* method proved to be the one with the worst trade-off performance among all tested methods. Although the other individual query strategies showed similar behavior, it was *STEP-GP:MaxVar* that showed a better overall trade-off performance compared to *STEP-GP:MaxEig*. In addition, the results obtained were consistent across 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 simplicity, 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.

8. Conclusion

Distributed optimization methods, such as ADMM, generally incur an excessive undesired communication overhead. In this context, the use of Gaussian processes has proven to be 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 reduction in communication depends in part on the mechanism through 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 by balancing two opposing criteria, which are to maximize the communication reduction while minimizing the error of the final solution obtained. Motivated by this constrained optimization problem and an alternative expression of the regular ADMM updates that showcases the inherent coupling between agents, we proposed a joint query strategy consisting in minimizing a convex communication cost restricted 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 whether the prediction is reliable enough to skip a communication round. The numerical results of solving a sharing problem with quadratic cost functions showed the different performances of the proposed methods in terms of the trade-off between reduction of communication cost and loss of accuracy in solving the optimization problem. In particular, the proposed joint query method achieved a better trade-off performance compared to the independent query strategies. Our next research steps include testing our proposed framework in more complex applications where we have more challenging objective functions, and convergence analysis of all query methods.

CRediT authorship contribution statement

Aldo Duarte: Software, Investigation, Data curation, Formal analysis, Writing – original draft. **Truong X. Nghiem:** Conceptualization, Software, Writing – review & editing, Visualization, Funding acquisition. **Shuangqing Wei:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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

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

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

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

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

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

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

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

and by inserting the definition of z_i^k we get

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

Taking the average of the definition of z_i^k we get $\bar{z}^k = \bar{y}^k - u^k$, and by inserting it into 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

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

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 2, 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

Consider the condition in (10). We introduce a unitary transformation U, whose columns are normalized eigenvectors of Σ_F, i.e., $\Sigma_F = U\Lambda U^\top$, where Λ is the diagonal matrix whose diagonal entries are the eigenvalues of Σ_F sorted in descending order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p > 0$. Given $F \sim \mathcal{N}(\mu, \Sigma_F)$, define $G = U^\top(F - \mu)$, which follows $\mathcal{N}(0, \Lambda)$. Moreover, $\|G\|_2 = \|F - \mu\|_2$. Consequently,

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

Let us define $Z_l = \frac{G_l}{\sqrt{\lambda_l}}$ for $1 \leq l \leq p$, with $Z_l \sim \mathcal{N}(0,1)$. Then, (B.1) can be expressed in terms of Z as

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

requiring the probability of being outside of an error sphere to be small.

Let $R = \sum_{l=1}^p \lambda_l Z_l^2$, which follows a weighted chi-square distribution, and $X = R - \sum_{l=1}^p \lambda_l$, we transform (B.2) as

$$\mathrm{P}\left[X + \sum_{l=1}^{p}\lambda_l \geq \|\mu\|_2^2\delta^2\right] \leq \xi. \tag{B.3}$$

We will follow the proof of Lemma 1 in [28] to get a bound for the inequality in (B.3). 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

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

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

$$\begin{aligned}\log[\mathbb{E}[\exp(uY)]] &= \sum_{l=1}^{p}\log\left[\mathbb{E}[\exp(ua_l(Z_l^2 - 1))]\right] \\ &\leq \sum_{l=1}^{p}\frac{a_l^2u^2}{1 - 2a_lu},\end{aligned} \tag{B.4}$$

which leads to the inequality

$$\log[\mathbb{E}[\exp(uY)]] \leq \frac{\|a\|_2^2u^2}{1 - 2\|a\|_\infty u}. \tag{B.5}$$

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

$$\log[\mathbb{E}[\exp(uY)]] \leq \frac{vu^2}{2(1 - 2cu)}, \tag{B.6}$$

then, for any positive x,

$$\mathrm{P}(Y \geq cx + \sqrt{2vx}) \leq \exp(-x). \tag{B.7}$$

Thus, given (B.5) and (B.6) we get $v/2 = \|a\|_2^2$ and $c = 2\|a\|_\infty$, which allow us to rewrite (B.7) as

$$\mathrm{P}(Y \geq 2\|a\|_\infty x + 2\|a\|_2\sqrt{x}) \leq \exp(-x). \tag{B.8}$$

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

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

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

Solving the quadratic equation we get that

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

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

$$x_{(w)} = \frac{\beta^2}{2\alpha^2} - \frac{\beta}{2\alpha^2}\sqrt{\beta^2 + 4\alpha w} + \frac{w}{\alpha}. \tag{B.9}$$

Introducing the definition of α and β into (B.9) we get

$$x_{(w)} = \frac{\|a\|_2^2}{2\|a\|_\infty^2} - \frac{\|a\|_2^2}{2\|a\|_\infty^2}\sqrt{1 + \frac{2w\|a\|_\infty}{\|a\|_2^2}} + \frac{w}{2\|a\|_\infty}, \tag{B.10}$$

which after some algebraic manipulations can be expressed as

$$x_{(w)} = \left(\sqrt{\frac{w}{2\|a\|_\infty} + \frac{\|a\|_2^2}{4\|a\|_\infty^2}} - \frac{\|a\|_2}{2\|a\|_\infty}\right)^2. \tag{B.11}$$

Inserting (B.11) and $\alpha x + \beta\sqrt{x} = w$ into (B.8), we get the expression for the desired probability as

$$\mathrm{P}[Y \geq w] \leq \exp(-x_{(w)}), \forall w \geq 0. \tag{B.12}$$

Going back to the context of the inequality in (B.3) given by

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

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

$$\mathrm{P}\left[X \geq \|\mu\|_2^2\delta^2 - \mathrm{tr}(\Sigma_F)\right] \leq \xi. \tag{B.13}$$

This probability can be also bounded following (B.12) as

$$\mathrm{P}\left[X \geq \|\mu\|_2^2\delta^2 - \mathrm{tr}(\Sigma_F)\right] \leq \exp(-x^*_{(\|\mu\|_2^2\delta^2 - \mathrm{tr}(\Sigma_F))}) \leq \xi, \tag{B.14}$$

where $x^*_{(\|\mu\|_2^2\delta^2 - \mathrm{tr}(\Sigma_F))}$ is the specific form for our problem of (B.11) which is defined as

$$x^*_{(\|\mu\|_2^2\delta^2 - \mathrm{tr}(\Sigma_F))} = \left(\sqrt{\frac{\|\mu\|_2^2\delta^2 - \mathrm{tr}(\Sigma_F)}{2\lambda_1} + \frac{\sum_{l=1}^{p}\lambda_l^2}{4\lambda_1^2}} - \frac{\sqrt{\sum_{l=1}^{p}\lambda_l^2}}{2\lambda_1}\right)^2, \tag{B.15}$$

with λ_l representing the eigenvalues of the covariance matrix Σ_F and λ_1 representing the biggest of those eigenvalues. Combining (B.14) and (B.15) we find a bound on the trace of Σ_F given by

$$-\left(\sqrt{\frac{\|\mu\|_2^2\delta^2 - \mathrm{tr}(\Sigma_F)}{2\lambda_1} + \frac{\sum_{l=1}^{p}\lambda_l^2}{4\lambda_1^2}} - \frac{\sqrt{\sum_{l=1}^{p}\lambda_l^2}}{2\lambda_1}\right)^2 \leq \ln(\xi)$$

$$\sqrt{\frac{\|\mu\|_2^2\delta^2 - \mathrm{tr}(\Sigma_F)}{2\lambda_1} + \frac{\sum_{l=1}^{p}\lambda_l^2}{4\lambda_1^2}} - \frac{\sqrt{\sum_{l=1}^{p}\lambda_l^2}}{2\lambda_1} \geq \sqrt{\ln(1/\xi)}$$

$$\frac{\|\mu\|_2^2\delta^2 - \mathrm{tr}(\Sigma_F)}{2\lambda_1} + \frac{\sum_{l=1}^{p}\lambda_l^2}{4\lambda_1^2} \geq \left(\sqrt{\ln(1/\xi)} + \frac{\sqrt{\sum_{l=1}^{p}\lambda_l^2}}{2\lambda_1}\right)^2$$

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

References

[1] N. Parikh, S. Boyd, Proximal algorithms, Found. Trends Optim. 1 (3) (2014) 127–239.

[2] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, K.H. Johansson, A survey of distributed optimization, Annu. Rev. Control 47 (2019) 278–305, http://dx.doi.org/10.1016/j.arcontrol.2019.05.006.

[3] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Comput. Math. Appl. 2 (1) (1976) 17–40, http://dx.doi.org/10.1016/0898-1221(76)90003-1.

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

[5] S. Kumar, R. Jain, K. Rajawat, Asynchronous optimization over heterogeneous networks via consensus ADMM, IEEE Trans. Signal Inf. Process. Netw. 3 (1) (2017) 114–129, http://dx.doi.org/10.1109/TSIPN.2016.2593896.

[6] X. Cao, K.J.R. Liu, Dynamic sharing through the ADMM, IEEE Trans. Automat. Control 65 (5) (2020) 2215–2222, http://dx.doi.org/10.1109/TAC.2019.2940317.

[7] Z. Liu, P. Dai, H. Xing, Z. Yu, W. Zhang, A distributed algorithm for task offloading in vehicular networks with hybrid fog/cloud computing, IEEE Trans. Syst., Man, Cybern.: Syst. (2021) 1–14, http://dx.doi.org/10.1109/TSMC.2021.3097005.

[8] T. Song, D. Li, Q. Jin, K. Hirasawa, Sparse proximal reinforcement learning via nested optimization, IEEE Trans. Syst., Man, Cybern.: Syst. 50 (11) (2020) 4020–4032, http://dx.doi.org/10.1109/TSMC.2018.2865505.

[9] R. Zhao, M. Miao, J. Lu, Y. Wang, D. Li, Formation control of multiple underwater robots based on ADMM distributed model predictive control, Ocean Eng. 257 (2022) 111585.

[10] P. Braun, L. Grüne, C.M. Kellett, S.R. Weller, K. Worthmann, A distributed optimization algorithm for the predictive control of smart grids, IEEE Trans. Automat. Control 61 (12) (2016) 3898–3911, http://dx.doi.org/10.1109/TAC.2016.2525808.

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

[12] C. Ma, J. Konecný, M. Jaggi, V. Smith, M.I. Jordan, P. Richtárik, M. Takác, Distributed optimization with arbitrary local solvers, Optim. Methods Softw. 32 (4) (2017) 813–848.

[13] S. Zhou, G.Y. Li, Communication-efficient ADMM-based federated learning, 2021, arXiv e-prints arXiv:2110.15318.

[14] W. Li, Y. Liu, Z. Tian, Q. Ling, Communication-censored linearized ADMM for decentralized consensus optimization, IEEE Trans. Signal Inf. Process. Netw. 6 (2020) 18–34, http://dx.doi.org/10.1109/TSIPN.2019.2957719.

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

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

[18] T.X. Nghiem, G. Stathopoulos, C. Jones, Learning proximal operators with Gaussian processes, in: Annual Allerton Conference on Communication, Control, and Computing, Illinois, USA, 2018.

[19] T.X. Nghiem, A. Duarte, 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, http://dx.doi.org/10.1109/IEEECONF51394.2020.9443553.

[20] A. Duarte, T.X. Nghiem, S. Wei, Communication-efficient ADMM using quantization-aware Gaussian process regression, 2022, http://dx.doi.org/10.36227/techrxiv.20448222.v1.

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

[22] Y. Xie, 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, http://dx.doi.org/10.1109/WSC.2016.7822135.

[23] C. Grigo, P.-S. Koutsourelakis, Bayesian model and dimension reduction for uncertainty propagation: Applications in random media, SIAM/ASA J. Uncertain. Quantif. 7 (1) (2019) 292–323, http://dx.doi.org/10.1137/17m1155867.

[24] H. Nagao, M. Srivastava, Fixed Width Confidence Region for the mean of a multivariate normal distribution, J. Multivariate Anal. 81 (2002) 259–273, http://dx.doi.org/10.1006/jmva.2001.2006.

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

[26] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, A. Vehtari, GPstuff: Bayesian modeling with Gaussian processes, J. Mach. Learn. Res. 14 (2013) 1175–1179.

[27] N.A. Nagendra, IEEE 802.11 MAC PROTOCOL, 2013, URL https://www.mathworks.com/matlabcentral/fileexchange/44110-ieee-802-11-mac-protocol.

[28] B. Laurent, P. Massart, Adaptive estimation of a quadratic functional by model selection, Ann. Statist. 28 (5) (2000) 1302–1338, URL http://www.jstor.org/stable/2674095.

[29] L. Birgé, P. Massart, Minimum contrast estimators on sieves: Exponential bounds and rates of convergence, Bernoulli 4 (3) (1998) 329–375.