Multi-agent Black-box Optimization using a Bayesian Approach to Alternating Direction Method of Multipliers ⋆

Dinesh Krishnamoorthy[1] **Joel A. Paulson**[2]

[1] *Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, 5600MB, The Netherlands.*
[2] *Department of Chemical and Biological Engineering, The Ohio State University, Columbus, OH 43210, USA.*
(email: d.krishnamoorthy@tue.nl, paulson.82@osu.edu)

Abstract: Bayesian optimization (BO) is a powerful black-box optimization framework that looks to efficiently learn the global optimum of an unknown system by systematically trading-off between exploration and exploitation. However, the use of BO as a tool for coordinated decision-making in multi-agent systems with unknown structure has not been widely studied. This paper investigates a black-box optimization problem over a multi-agent network coupled via shared variables or constraints, where each subproblem is formulated as a BO that uses only its local data. The proposed multi-agent BO (MABO) framework adds a penalty term to traditional BO acquisition functions to account for coupling between the subsystems without data sharing. We derive a suitable form for this penalty term using alternating directions method of multipliers (ADMM), which enables the local decision-making problems to be solved in parallel (and potentially asynchronously). The effectiveness of the proposed MABO method is demonstrated on an intelligent transport system for fuel efficient vehicle platooning.

Keywords: Distributed optimization, Bayesian Optimization, Multi-agent systems

1. INTRODUCTION

Multi-agent decision-making problems (often formulated as optimization tasks) arise in a wide range of application areas, such as in collaborative robotics, intelligent transport systems, integrated energy systems, sensor networks, and modular manufacturing, where a collection of agents work together to achieve one or more well-defined goals. A common framework for multi-agent decision-making is the decomposition-coordination approach, where the agents solve their own local subproblems, and a central coordinator accounts for the coupling/interactions between the different agents, thus facilitating a coordinated action (Bertsekas, 2016; Lasdon, 2002). One such popular distributed optimization approach is the alternating directions method of multipliers (ADMM) Boyd et al. (2011).

In the traditional optimization framework, each local agent requires analytical expressions (i.e., models) that relate the local decision variables to the local objectives and constraints in every subsystem. Obtaining reliable models is a time consuming and tedious work, and modeling effort is often one of the biggest bottlenecks in many application areas (Chachuat et al., 2009). Accurate models may also be challenging to obtain in some applications, due to lack of domain knowledge. This requirement hinders the use of traditional distributed optimization methods that rely on exploiting the structure of such models (in the form of gradients or relaxations) on applications for which accurate models are not readily available.

A class of optimization methods known as "black box" optimization is a promising alternative that circumvents the need for detailed mathematical models. Here, the system is considered to be an unknown "black-box" that can be queried through a series of simulations or experiments. The data observed from such evaluations can then be used to find the optimum. Bayesian optimization (BO) is one such zeroth-order black-box optimization scheme, where the optimum is learned by sequentially interacting with the system (Kushner, 1964; Shahriari et al., 2015; Jones et al., 1998).

However, in the context of multi-agent systems, formulating the subproblems using BO does not account for the coupling and the interactions between the different agents. In BO, the next sequence of actions are computed by optimizing a so-called "acquisition function" induced from a probabilistic surrogate model for the local objective function. Several different acquisition functions have been proposed in the BO literature, which fundamentally attempt to trade-off exploration (learning for the future) and exploitation (immediately advancing toward goal). Interested readers are referred to the review papers (Shahriari et al., 2015; Frazier, 2018) for detailed discussions on the different type of acquisition functions and the relationship between them. In the distributed setting, however, the local acquisition function in each

⋆ D.K. acknowledges support from the Research Council of Norway through the IKTPLUSS program and Eindhoven University of Technology. J.A.P. acknowledges support from the National Science Foundation Grant 2036982.

subsystem does not account for the coupling to the other subsystems. A truly distributed BO algorithm requires the local acquisition functions to be modified so that, e.g., a central coordinator can influence the local BO decision-making in each subsystem. Despite the growing popularity of BO, decomposition-coordination frameworks for BO has received very limited attention in the literature.

To address this gap, this paper proposes a multi-agent Bayesian optimization (MABO) framework that is capable of solving black-box optimization problems defined over a multi-agent network coupled via shared variables or constraints. An important and realistic assumption made about the network is that direct data sharing is not allowed between the local agents due to data privacy concerns or bandwidth limitations. Such constraints would not be satisfied by traditional "centralized" BO methods, but is naturally handled by our proposed MABO approach, which augments the local acquisition functions with additional penalty terms expressed directly in terms of limited information received from the central coordinator. MABO can be used in combination with any existing acquisition function from the BO literature and, thus, provides a bridge between the developments in BO and distributed optimization.

The reminder of the paper is organized as follows. Section 2 introduces the preliminaries of Bayesian optimization. Section 3 describes the proposed MABO framework and corresponding penalty terms that are derived using ADMM. The effectiveness of the MABO framework is then demonstrated on a fuel efficient vehicle platooning problem in Section 4, wherein the goal is to minimize overall fuel consumption of the platoon by selecting an optimal cruising speed. Lastly, we conclude the paper and discuss some interesting directions for future work in Section 5.

2. PRELIMINARIES - BAYESIAN OPTIMIZATION

Consider the optimization problem

$$\min_{x \in \mathcal{X}} f(x) \tag{1}$$

where $x \in \mathcal{X}$ is the decision variable chosen from a known compact set $\mathcal{X}$ and $f : \mathcal{X} \to \mathbb{R}$ is some cost black-box function whose mathematical structure is unknown. Bayesian optimization (BO) consists of the following two steps:

(1) **Learning**: A probabilistic surrogate model, typically a Gaussian process (GP), is derived for the unknown cost function $f(x)$. The prior distribution for the surrogate model is updated every time that a noisy observation of the cost $y = f(x) + w$, where w is a stochastic noise term, is available by conditioning on the observations (Rasmussen and Williams, 2006).

(2) **Decision-making:** The posterior surrogate model is then used to induce a suitable acquisition function $\alpha : \mathcal{X} \to \mathbb{R}$, which provides a measure of how desirable querying a point $x \in \mathcal{X}$ is expected to be with respect to optimizing the cost. Using this, the next action to evaluate is computed by solving [1]

[1] Note that we choose to define the acquisition function $\alpha(\cdot)$ in a way that its a minimization problem (as opposed to maximization problem) without loss of generality.

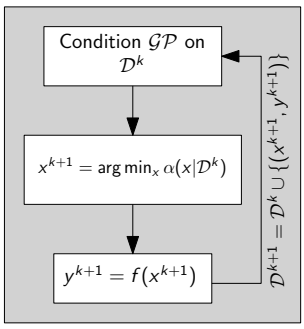

Fig. 1. Illustration of Bayesian optimization scheme to minimize an unknown function $f(x)$.

$$x^{k+1} = \arg\min_{x \in \mathcal{X}} \alpha(x|\mathcal{D}^k), \tag{2}$$

where $\mathcal{D}^k := \{(x^j, y^j)\}_{j=1}^k$ is the past set of observations at time k and $\alpha(x|\mathcal{D}^k)$ denotes the acquisition function evaluated at x induced by the posterior conditioned on $\mathcal{D}_k$.

The choice of acquisition function controls the trade-off between exploration and exploitation when selecting a suitable next action, with the goal being to find the global optimum in as few iterations as possible. There are several choices for the acquisition function such as probability of improvement, expected improvement, upper confidence bound, and Thompson sampling. See Shahriari et al. (2015); Frazier (2018) for a more detailed description of the different acquisition functions.

In summary, BO is a black-box decision-making strategy where we sequentially query the system, observe the cost, update our prior belief and choose the next action to query. The BO loop is schematically represented in Fig. 1.

3. PROPOSED MULTI-AGENT BAYESIAN OPTIMIZATION (MABO) FRAMEWORK

Now, let us consider a multi-agent version of (1) that has the following additively separable cost function

$$\min_{x \in \mathcal{X}} f(x) = \textstyle\sum_{i=1}^N f_i(x), \tag{3}$$

where $f_i : \mathcal{X} \to \mathbb{R}$ denotes a local cost function for the ith subsystem. We make the following assumption about these local cost functions.

Assumption 1. The local cost functions $f_i(x)$ are unknown but can be evaluated at any desired $x \in \mathcal{X}$ for all $i = 1, \ldots, N$.

If this was our only assumption, we could apply traditional BO methods (Fig. 1) on the aggregated set of data from all subsystems. It is worth noting that an improved version of this algorithm can easily be developed by exploiting the additive cost structure in (3), as discussed in Kudva et al. (2022). However, such methods are centralized in the sense that they require the cost data to be gathered from all local subsystems at every iteration. There are many situations where this is not desired or even possible, which we summarize in the following additional assumption.

Assumption 2. Data cannot be shared across subsystems such that subsystem i cannot directly use data from any other subsystem $j \neq i$.

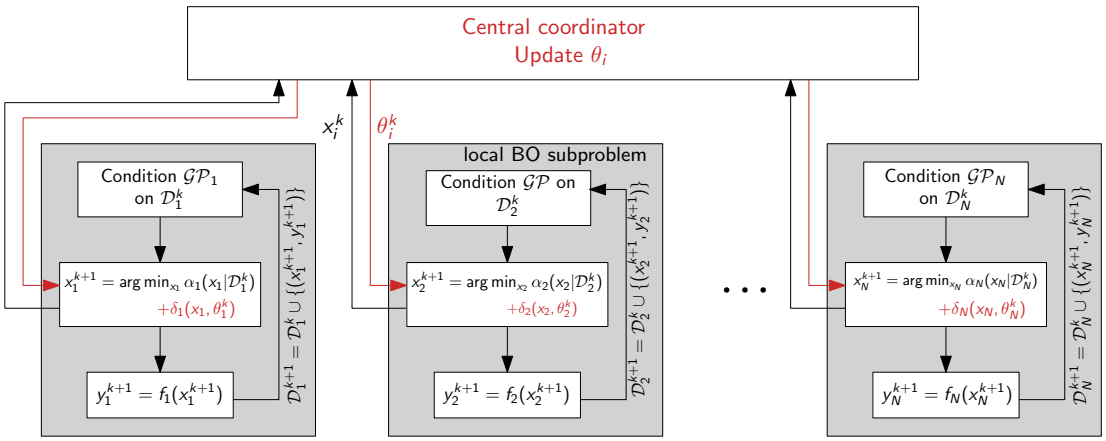

Fig. 2. Multi-agent Bayesian optimization (MABO) scheme where each subproblem minimizes its unknown local cost function $f_i(x_i)$ using a modified BO method. Each subproblem learns only from the local data set $\mathcal{D}_i$ which is not shared among the different subsystems. The acquisition function in each subproblem has additional penalty terms as a function of the master variables θ_i that enables coordinated action of the different subsystems.

A straightforward approach to tackle (3), under Assumptions 1–2, is to separately apply BO to each subproblem. That is, learn a GP surrogate model for each local cost function $f_i(x_i)$ based only on its local observations $\mathcal{D}_i^k = \{(x_i^j, y_i^j)\}_{j=1}^k$. Any suitable acquisition function $\alpha_i(x_i|\mathcal{D}_i^k)$ induced by the GP posterior is then used for local decision-making by each agent. However, computing the next action by optimizing only the acquisition function induced locally

$$x_i^{k+1} = \arg\min_{x_i \in \mathcal{X}} \ \alpha_i(x_i|\mathcal{D}_i^k),$$

results in each of the different subsystems taking independent decisions without any coordination. Solving the multi-agent version of the problem (3) requires a coordinated solution in which $x_1^\star = \cdots = x_N^\star$, which is not necessarily achieved with the aforementioned strategy.

To enable the required coordination, we propose to incorporate additional penalty terms to the local acquisition function. That is, each subsystem now computes its next action x_i^{k+1} by optimizing

$$x_i^{k+1} = \arg\min_{x_i \in \mathcal{X}} \ \alpha_i(x_i|\mathcal{D}_i^k) + \delta_i(x_i, \theta_i^k), \tag{4}$$

where $\theta_i^k \in \mathbb{R}^{n_\theta}$ denotes the information that is communicated to the ith subsystem at the kth iteration (e.g. by a central coordinator), $x_i \in \mathcal{X}$ denotes the local decision variables, $\alpha_i(\cdot)$ is any suitable acquisition function used to optimize the unknown local cost $f_i(\cdot)$, and $\delta_i : \mathcal{X} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$ is the augmented penalty term that can influence the local decisions. We propose to derive an expression for $\delta_i(\cdot)$ using the alternating direction method of multipliers (ADMM) framework, as described below.

The optimization problem (3) can be equivalently rewritten by introducing a new variable x_0 as

$$\min_{x_0, \{x_i\} \in \mathcal{X}} \ \textstyle\sum_{i=1}^N f_i(x_i) \text{ s.t. } x_i = x_0 \quad \forall i = 1, \ldots, N. \tag{5}$$

The augmented Lagrangian for (5) is given by

$$\min_{x_0, \{x_i\} \in \mathcal{X}} \ \sum_{i=1}^N \left[f_i(x_i) + \lambda_i^\mathsf{T}(x_i - x_0) + \frac{\rho}{2}\|x_i - x_0\|^2 \right], \tag{6}$$

where $\lambda = (\lambda_1, \ldots, \lambda_N)$ denotes the set of dual variables (or Lagrange multipliers) associated with the equality consraints in (5) and $\rho > 0$ is a penalty parameter. The ADMM scheme, which is derived from the augmented Lagrangian in (6), then consists of the following iterations (Boyd et al., 2011, Chapter 7)

$$x_0^{k+1} = \frac{1}{N} \sum_{i=1}^N \left[x_i^k + \frac{\lambda_i^k}{\rho} \right], \tag{7a}$$

$$x_i^{k+1} = \arg\min_{x_i \in \mathcal{X}} f_i(x_i) + \lambda_i^{k\mathsf{T}} x_i + \frac{\rho}{2}\|x_i - x_0^{k+1}\|^2, \tag{7b}$$

$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - x_0^{k+1}), \tag{7c}$$

Since $f_i(x_i)$ in each subproblem (7b) is unknown under Assumption 1, we propose to replace it with a local acquisition function $\alpha_i(x_i|\mathcal{D}_i^k)$. In this case, (7b) has the same form as (4) with the penalty term equal to

$$\delta_i(x_i, \theta_i^k) = \lambda_i^{k\mathsf{T}} x_i + \frac{\rho}{2}\|x_i - x_0^{k+1}\|^2, \tag{8}$$

where $\theta_i^k := (x_0^{k+1}, \lambda_i^k)$ denotes the coordinating variables, which are updated as shown in (7a) and (7c) at every iteration; these updates are very simple, can be stored with little memory, and ensure that Assumption 2 is satisfied. We refer to this overall proposed framework as multi-agent BO (MABO), which is illustrated in Fig. 2. We also provide a complete algorithmic description of MABO in Algorithm 1, which has several unique features that we highlight in the series of remarks below.

Remark 1. (Data privacy). The GP model in each subsystem is conditioned only on the local cost measurement, which need not be shared with the other subsystems. This enables a collection of agents to take coordinated actions without sharing its local data with one another, clearly respecting data privacy concerns or any other data sharing limitations (such as bandwidth or connection limitations).

Remark 2. (Choice of acquisition function). Note that the choice of the local acquisition function is independent from the coordinating term $\delta_i(\cdot)$. Hence any acquisition function proposed in the standard BO framework can be directly used by our MABO approach. Moreover, the choice of the acquisition function can also be different between the different subsystems. Simply put, in a multi-agent network, each agent is free to choose its local acquisition function, without affecting the structure of the MABO strategy.

Algorithm 1 Multi-agent Bayesian optimization.

Input: N local GP models for the local cost functions $f_i(x_i)$, initial values x_i^0 and λ_i^0, and penalty $\rho > 0$

for ADMM iterates $k = 0, 1, 2, \ldots$ **do**
  $x_0^{k+1} \leftarrow \frac{1}{N}\sum_{i=1}^{N}\left[x_i^k + \frac{\lambda_i^k}{\rho}\right]$ ▷ Central collector
  for subproblems $i = 1, \ldots, N$ **do** (in parallel)
    $\theta_i^k \leftarrow [x_0^{k+1}, \lambda_i^k]^\mathsf{T}$
    $\delta_i(x_i, \theta_i^k) \leftarrow \lambda_i^{k\mathsf{T}} x_i + \frac{\rho}{2}\|x_i - x_0^{k+1}\|^2$
    for local BO iterates $j = 1, 2, \ldots$ **do**
      Update local GP model using $\mathcal{D}_i^k$
      Induce any acquisition function $\alpha_i(x_i|\mathcal{D}_i^k)$
      $x_i^{k+1,j} \leftarrow \arg\min_{x_i \in \mathcal{X}} \alpha_i(x_i|\mathcal{D}_i^k) + \delta_i(x_i, \theta_i^k)$
      Query and observe the local cost $f_i(x_i^{k+1,j})$
      $\mathcal{D}_i^k \leftarrow \mathcal{D}_i^k \cup \{(x_i^{k+1,j}, f_i(x_i^{k+1,j}))\}$
    end for
    $\lambda_i^{k+1} \leftarrow \lambda_i^k + \rho(x_i^{k+1,j} - x_0^{k+1})$
  end for
end for

Output: Optimal solution x_0

Remark 3. (Multiple BO iterations). It can be seen that our proposed decomposable MABO framework (Algorithm 1) has a hierarchical structure with two levels of iterations, namely, the local BO iterations within each subproblem, and the ADMM iterations between the central coordinator and the subproblems. As such, MABO can also be implemented with several BO iterations per ADMM iteration. For example, after every ADMM iterate that updates θ_i^k, the local subproblems can run one or several local Bayesian optimization iterates for a given θ_i^k. Furthermore, the different subproblems can also run different number of BO iterations for each ADMM iterate. This is especially beneficial where the local evaluation time and budgets are different in the different agents. This is explicitly shown in Algorithm 1, where the local BO iterations j can be run for any desired finite number of steps.

Remark 4. (Optimal sharing). Although we have focused our MABO derivation on the consensus problem in (5), the approach extends to *optimal sharing problems* that have coupling constraints of the form $\sum_i A_i x_i = 0$. The main modification needed is to update the form of the penalty term $\delta_i(\cdot)$ whose structure is shown in Boyd et al. (2011, Section 7.3). Many important applications involving the distribution of a shared resource would fall into this class of problems (e.g., allocation of energy from a centralized repository in a chemical process).

Remark 5. Although we considered only the ADMM framework in this section, (4) also allows for other decomposition strategies such as primal decomposition and Lagrangian decomposition, with $\delta_i(\cdot)$ and θ_i chosen accordingly.

4. ILLUSTRATIVE EXAMPLE: FUEL EFFICIENT VEHICLE PLATOONING

We demonstrate the performance of MABO on the problem of fuel-efficient vehicle platooning, which is a multi-agent system consisting of several vehicles that take coordinated action to optimize the overall fuel efficiency. Here, "platooning" refers to a collection of vehicles that drive in a single-file fashion at close inter-vehicle distance in highways. Platooning of heavy-duty vehicles (e.g., trucks used for freight transport) has several benefits such as reduced traffic congestion, greater fuel economy due to reduced air drag friction, and fewer traffic collisions (Franke et al., 1995; Browand et al., 2004).

Early work on truck platooning focused on tracking inter-vehicle distances using adaptive cruise control (Hao and Barooah, 2013); however, there has been an increasing interest in optimizing the speed of the platoon to minimize overall fuel consumption. In fact, a recent experimental study showed that, although the fuel economy improved at all platooning speeds, certain speeds resulted in the best overall fuel economy (Lammert et al., 2014). However, the effect of cruising speed on fuel consumption is not constant and depends on several factors such as freight load, road and driving conditions, time since last engine maintenance, etc., to name a few. Therefore, developing accurate models that can be used to find the optimal platoon speed is challenging.

An attractive alternative is to resort to a black-box optimization strategy. Black-box formulations are particularly useful when the platoon consists of vehicles that have different configurations and/or manufacturers since it is non-obvious how to coordinate different types of vehicles to minimize overall fuel consumption when each isolated vehicle has a unique optimum speed. Such problems can be formulated in the form of (3) where x_i and $f_i(x_i)$, respectively, denote the speed and fuel consumption of the ith vehicle. Here, we simulate the fuel consumption of a platoon of $N = 7$ vehicles using the approach derived by Sobrino et al. (2016), which leads to a model of the form

$$f_i(x_i) = a_i + \frac{b_i}{x_i} + c_i x_i + d_i x_i^2, \tag{9}$$

where speed x_i has units of [km/h] and fuel consumption $f_i(x_i)$ has units of [g/veh-km]. The nominal model parameters are given by the experimentally calibrated values in (Sobrino et al., 2016, Section 5). To emulate $N = 7$ completely different trucks, we sample perturbed model parameters from a uniform distribution defined by $\pm 20\%$ of these nominal values. We highlight that (9) is used only as a source of data generation, as the parameter values and structure are unknown in accordance with Assumption 1.

The proposed MABO approach (Algorithm 1) was implemented in `Python` using the `GPy` package (GPy, since 2012) to construct and train the GP models using a squared exponential (SE) kernel. We first use a lower confidence bound (LCB) acquisition function defined as follows

$$\alpha_i(x_i|\mathcal{D}_i^k) := \mu_i^k(x_i) - \sqrt{\beta}\sigma_i^k(x_i), \tag{10}$$

where μ_i^k and σ_i^k, respectively, denote the posterior mean and standard deviation predicted by the GP model for the ith subsystem at the kth iteration and β is a scaling parameter used to trade-off between exploration and exploitation. Based on previous work, we select $\beta = 4$ throughout this work (Srinivas et al., 2009). We selected a penalty value of $\rho = 10$ and only run $j = 1$ local BO iteration for every ADMM iteration.

We first demonstrate the effect of the coordination term in Fig. 3, which plots the GP model and corresponding

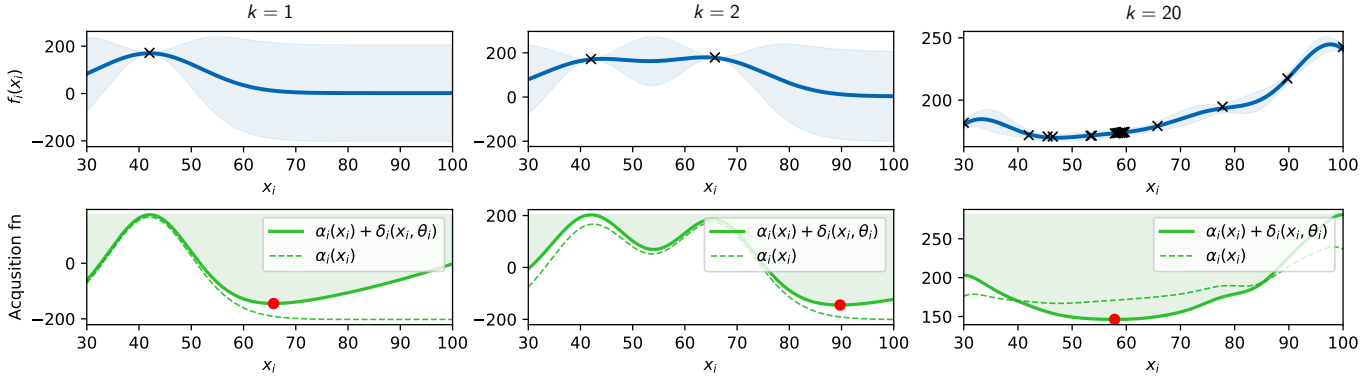

Fig. 3. The local Bayesian optimization of one of the subsystems at iterates $k = 1, 2$, and 20. Top subplots show the Gaussian process of the local cost. Bottom subplots show the acquisition function with (solid green) and without (dashed green) coordination. The next query point is shown as a red dot.

acquisition functions for a single vehicle at three iterations $k \in \{1, 2, 20\}$. From the bottom subplots, we see that the local non-perturbed acquisition function $\alpha_i(x_i|\mathcal{D}_i^k)$ (green dashed lines) behaves quite different from the MABO acquisition $\alpha_i(x_i|\mathcal{D}_i^k)+\delta_i(x_i, \theta_i^k)$ (green solid line). Already at iteration $k = 1$, the effect of the penalty term $\delta_i(x_i, \theta_i^k)$ is evident, which skews the selection process toward lower x_i^{k+1} values such that the next query point is not too far from the average speed of the platoon. By iteration $k = 20$, the local BO step converges to around 57 km/hr (near the true optimal value), which significantly differs from the optimum point of the unconstrained acquisition function.

We repeat the same simulation test using two additional types of acquisition functions, probability of improvement (PI) and expected improvement (EI), in addition to LCB. Furthermore, to demonstrate the flexibility of MABO, we also test a "mixed" case wherein the different vehicles use different acquisition functions (i.e., vehicles 1 and 2 use EI, vehicles 3 and 4 use PI, vehicles 5 and 6 use LCB, and vehicle 7 using a greedy mean-based policy). The individual vehicle speeds obtained at each MABO iteration using the different acquisition function test cases are shown in Fig. 4. We also show the true optimum with a black dashed line to serve as a benchmark to compare against. The average speed of the platoon x_0^k, the primal residual $r^k := \sum_i \|x_i^k - x_0^k\|_2^2$, and the dual residual $s^k := N\rho^2\|x_0^k - x_0^{k-1}\|_2^2$ for the different cases are shown in Fig. 5 to compare the effect of the choice of the acquisition function on the progress of the ADMM iterations. It can be seen that the progress of the ADMM iterates are not heavily affected by the choice of the acquisition function. We also benchmark this against the traditional model-based ADMM with full model information in each subsystem, which acts as a lower bound on the achievable residuals. This is shown in solid black lines. This empirically quantifies the additional iterations needed to explore the unknown function when using BO in the subsystems.

5. CONCLUSION AND FUTURE WORK

This paper presents a novel approach for efficiently finding global optimum solutions of multi-agent systems with unknown (black-box) structure and limitations on information sharing by combining elements of distributed optimization with Bayesian optimization (BO). As opposed to independently applying BO in each of the subsystems, the key idea is to modify the local acquisition (or expected utility) function with additional penalty terms that are properly set according to a master coordinator, which requires limited communication bandwidth and memory storage. Our proposed multi-agent BO (MABO) framework is broadly applicable in the sense that it works for different network structures (e.g., consensus and sharing formulations) and allows for the use of any well-defined acquisition function within the local subproblems. Furthermore, the local subproblems can be straightforwardly formulated using constrained BO (Gelbart et al., 2014; Paulson and Lu, 2022) or safe BO methods (Berkenkamp et al., 2021; Krishnamoorthy and Doyle III, 2022) whenever local black-box constraints are present in a given subsystem. The proposed MABO approach is demonstrated on a vehicle platooning problem, where the goal is to select a common platoon speed for multiple (different configuration) vehicles such that the overall fuel consumption is minimized. We show that MABO is capable of quickly identifying the optimal platoon speed, even with no prior knowledge of the relationship between speed and fuel consumption in any of the vehicles in the platoon. This work represents a first step toward a decomposable BO framework and future work should involve better understanding the theoretical properties (e.g., rate of convergence) of the proposed method. In addition, it would be interesting to explore the impact of the update rule chosen for the central coordinator on performance, as there may exist strategies that accelerate the speed of convergence in certain applications.

REFERENCES

Berkenkamp, F., Krause, A., and Schoellig, A.P. (2021). Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 1–35.

Bertsekas, D.P. (2016). *Nonlinear Programming*. Athena Scientific, Belmont, MA, 3 edition.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1), 1–122.

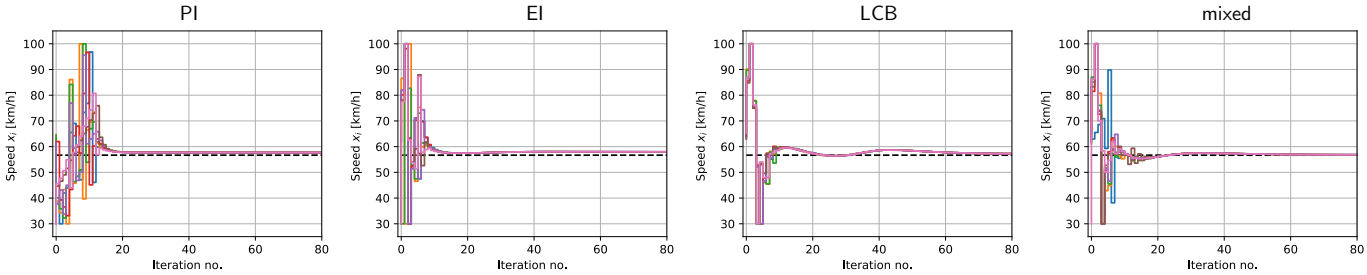

Fig. 4. Vehicle speeds when the local BO loops use probability of improvement (leftmost subplot), expected improvement (middle left subplot), LCB (middle right subplot), and mixed (rightmost subplot) acquisition functions. The dashed black line represents the true solution.

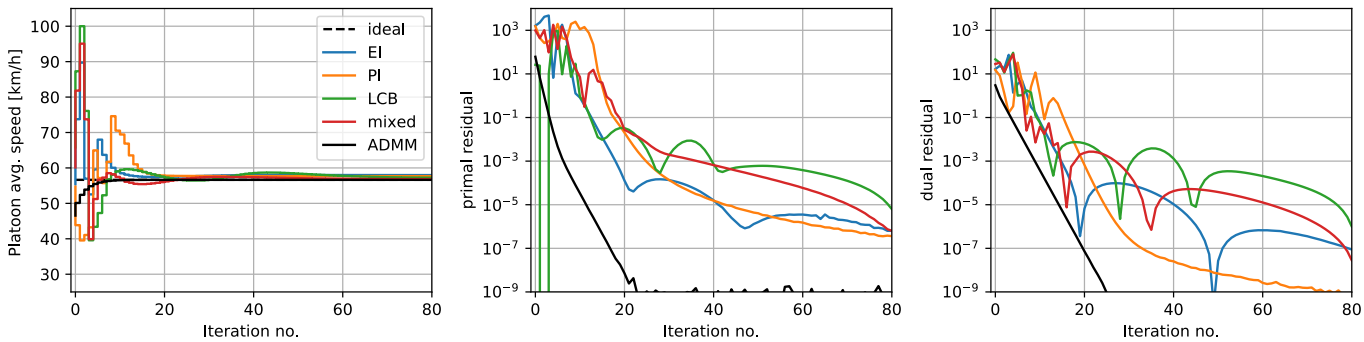

Fig. 5. Comparison of platoon average speed, primal residual, and dual residual when local BO loops use probability of improvement (orange), expected improvement (blue), LCB (green), and mixed (red) acquisition functions. The solid black lines represent the ideal ADMM iterations when the local functions are perfectly known.

Browand, F., McArthur, J., and Radovich, C. (2004). Fuel saving achieved in the field test of two tandem trucks.

Chachuat, B., Srinivasan, B., and Bonvin, D. (2009). Adaptation strategies for real-time optimization. *Computers & Chemical Engineering*, 33(10), 1557–1567.

Franke, U., Bottiger, F., Zomotor, Z., and Seeberger, D. (1995). Truck platooning in mixed traffic. In *Proceedings of the Intelligent Vehicles' 95. Symposium*, 1–6. IEEE.

Frazier, P.I. (2018). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Gelbart, M.A., Snoek, J., and Adams, R.P. (2014). Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, 250–259. AUAI Press.

GPy (since 2012). GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`.

Hao, H. and Barooah, P. (2013). Stability and robustness of large platoons of vehicles with double-integrator models and nearest neighbor interaction. *International Journal of Robust and Nonlinear Control*, 23(18), 2097–2122.

Jones, D.R., Schonlau, M., and Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4), 455–492.

Krishnamoorthy, D. and Doyle III, F.J. (2022). Safe Bayesian Optimization using Interior-Point Methods - Applied to Personalized Insulin Dose Guidance. *IEEE Control System Letters (In-Press)*.

Kudva, A., Sorouifar, F., and Paulson, J.A. (2022). Efficient robust global optimization for simulation-based problems using decomposed Gaussian processes: Application to MPC calibration. In *In Proceedings of the American Control Conference*, 2091–2097. IEEE.

Kushner, H.J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86, 97–106.

Lammert, M.P., Duran, A., Diez, J., Burton, K., and Nicholson, A. (2014). Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

Lasdon, L.S. (2002). *Optimization theory for large systems*. Courier Corporation.

Paulson, J.A. and Lu, C. (2022). COBALT: COnstrained Bayesian optimizAtion of computationaLly expensive grey-box models exploiting derivaTive information. *Computers & Chemical Engineering*, 160, 107700.

Rasmussen, C.E. and Williams, C.K. (2006). *Gaussian processes for machine learning*. MIT press Cambridge, MA.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., and De Freitas, N. (2015). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.

Sobrino, N., Monzon, A., and Hernandez, S. (2016). Reduced carbon and energy footprint in highway operations: the highway energy assessment (hera) methodology. *Networks and Spatial Economics*, 16(1), 395–414.

Srinivas, N., Krause, A., Kakade, S.M., and Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*.